RODRIGO AMADOR COELHO

ALGORITMO DE ENXAME DE PARTÍCULAS ENSEMBLE PARA CLUSTERIZAÇÃO DE DADOS

LAVRAS – MG

2014

RODRIGO AMADOR COELHO

ALGORITMO DE ENXAME DE PARTÍCULAS ENSEMBLE PARA CLUSTERIZAÇÃO DE DADOS

Dissertação apresentada à Universidade Federal de Lavras, como parte das exigências do Programa de Pós-Graduação em Ciência da Computação, área de concentração em Inteligência Computacional e Processamento Gráfico, para a obtenção do título de Mestre.

Orientador

Dr. Ahmed Ali Abdalla Esmin

LAVRAS – MG

2014

Ficha Catalográfica Elaborada pela Coordenadoria de Produtos e Serviços da Biblioteca Universitária da UFLA

Coelho, Rodrigo Amador.
Algoritmo de enxame de partículas ensemble para clusterização de dados / Rodrigo Amador Coelho. – Lavras : UFLA, 2014.
72 p. : il.

Dissertação (mestrado) – Universidade Federal de Lavras, 2014.
Orientador: Ahmed Ali Abdalla Esmin.
Bibliografia.

1. Particle Swarm Optimization. 2. Clusterização. 3. Ensemble. 4. Função de consenso. I. Universidade Federal de Lavras. II. Título.

CDD – 004.35

RODRIGO AMADOR COELHO

ALGORITMO DE ENXAME DE PARTÍCULAS ENSEMBLE PARA CLUSTERIZAÇÃO DE DADOS

Dissertação apresentada à Universidade Federal de Lavras, como parte das exigências do Programa de Pós-Graduação em Ciência da Computação, área de concentração em Inteligência Computacional e Processamento Gráfico, para a obtenção do título de Mestre.

APROVADA em 27 de fevereiro de 2014.

Dr. Ahmed Ali Abdalla Esmin	UFLA
Dr. Carlos H. Valério de Moraes	UNIFEI
Dr. Denilson Alves Pereira	UFLA

Dr. Ahmed Ali Abdalla Esmin
Orientador

LAVRAS – MG

2014

Aos meus pais, Salustriano e Gilca,
por sempre acreditarem que eu seria capaz.

DEDICO

AGRADECIMENTOS

Durante o tempo de realização deste Mestrado, muitas pessoas contribuíram em meus trabalhos das mais diferentes formas. A todos, os meus sinceros agradecimentos.

Agradeço à Universidade Federal de Lavras – UFLA, ao Departamento de Ciência da Computação – DCC e ao Programa de Pós- Graduação em Ciência da Computação da UFLA – PPGCC/UFLA, pela estrutura oferecida e pela oportunidade de realização do Mestrado.

Agradeço a Capes, CNPq e FAPEMIG -pela concessão da bolsa de estudos, que tornou possível a realização do Mestrado.

RESUMO

Clusterização é uma importante tarefa na mineração de dados e tem sido utilizada por muitos pesquisadores em diferentes áreas. O método do *ensemble* de *clusters* utiliza de vários resultados de diferentes algoritmos de clusterização em uma solução de consenso para melhorar a qualidade e solidez dos resultados. Geralmente construído de duas fases, o *ensemble* de *clusters*, em sua primeira fase é composto de um conjunto de algoritmos que recebe a base de dados e tem como saída um conjunto de *clusters* como solução. A segunda fase recebe o conjunto de *clusters* como entrada e as combina por meio de uma função de consenso produzindo *clusters* finais. Considerado uma alternativa robusta e precisa, frente a algoritmos individuais de clusterização, o *ensemble* de *clusters* melhora o resultado compensando a possibilidade de erros cometidos por alguns algoritmos de clusterização pela intervenção da solução correta de outros. Um dos maiores desafios, além da função de consenso, é determinar a melhor estrutura da base de dados que será usada pela função de consenso. Nesse trabalho, o algoritmo *Particle Swarm Optimization* (PSO) é proposto como algoritmo de clusterização para a primeira fase do *ensemble* e como função de consenso na segunda fase. Diferentes medidas de similaridade foram utilizadas, além de dois tipos de estruturas de base de dados, que servirão como entrada para a função de consenso. Foram realizadas três baterias de experimentos a fim de investigar o comportamento do PSO em um *ensemble* de *clusters*. Um dos experimentos realizados consiste na aplicação do PSO em um *ensemble* a fim de predizer defeitos em *software*. Ao fim do estudo empírico, o *ensemble* de *clusters* com o PSO foi capaz de produzir resultados tão bons ou melhores, nas duas diferentes estruturas de bases de dados.

Palavras-chave: *Particle Swarm Optimization.* Clusterização. *Ensemble.* Função de Consenso.

ABSTRACT

Clustering is an important task in data mining and has been used by many researchers in different areas. The cluster ensemble method uses several results of different clustering algorithms at a consensus solution to improve the quality and robustness of the results. Generally built in two phases, in the first stage the cluster ensemble is comprised of a set of algorithms that receive the database and has as output a set of clusters as a solution. The second stage receives the set of clusters as input and combines them through a consensus function producing final clusters. Considered a precise and robust alternative compared individual clustering algorithms, the clustering ensemble improves result using the possibility of compensating errors committed by some clustering algorithms for intervention of other correct solution. One of the major challenges beyond the consensus function is to determine the best structure of the data set that will be used by the function consensus. In this work, the Particle Swarm Optimization algorithm (PSO) is proposed as a clustering algorithm for the first phase of the ensemble and as a consensus function in the second phase. Different similarity measures and two types of database structures serve as input to the consensus function. Three sets of experiments were performed to investigate the behavior of PSO in a cluster ensemble. One of the experiments involves the application of PSO in an ensemble in order to predict defects in software quality. At the end of the empirical study, the PSO clustering ensemble was able to produce as good or better results even when using two different structures from databases.

Keywords: Particle Swarm Optimization. Clustering. Ensemble. Consensus Function.

LISTA DE FIGURA

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

1.1 Contextualização

Nos últimos anos, houve um crescimento na geração, aquisição e armazenamento de dados. Esta grande quantidade de dados armazenados contém valiosos e importantes conhecimentos ocultos, que poderiam ser utilizados para melhorar o processo de decisão de uma organização. A extração de informações e conhecimentos, considerando esta quantidade enorme e continuamente crescente de dados, tornou-se uma tarefa muito complexa e ultrapassou a capacidade humana de processar, analisar e compreender. Portanto, um processo para a conversão de grandes quantidades de dados para o conhecimento tornar-se inestimável (HAN; KAMBER, 2005).

A área de Knowledge Discovery in Databases (KDD), Descoberta de Conhecimento em Base de Dados, surgiu na última década para enfrentar este desafio. A mineração de dados é uma parte do processo de KDD. Tecnicamente, mineração de dados é o processo de extrair padrões úteis ou minerar conhecimento de grandes quantidades de dados por meio da aplicação de algoritmos apropriados, ferramentas e técnicas. Clustering é uma das mais importantes e bem conhecidas tarefas de mineração de dados (HAN; KAMBER, 2005; Tan; Steinbach; Kumar, 2005).

Clustering é o processo de agrupamento de um conjunto de dados não rotulados em grupos de objetos semelhantes. Cada grupo, chamado de cluster, consiste de objetos que são semelhantes entre si, no que diz respeito à certa medida de similaridade e que são diferentes de objetos de outros grupos. As aplicações de análise de cluster têm sido usadas em muitas diferentes áreas, incluindo inteligência artificial, bioinformática, biologia, visão computacional, compressão de dados, análise de imagens, recuperação de informação,

aprendizado de máquina, marketing, medicina, reconhecimento de padrões, análise de base de dados espacial, estatísticas, sistemas de recomendação e web mining (KRIEGEL; KRÖGER; ZIMEK, 2009; LUXBURG, 2007).

O método de ensemble de clusters tem sido utilizado para melhorar a estabilidade e robustez das saídas da clusterização (TOPCHY; JAIN; PUNCH, 2004). Nesse método, vários algoritmos fornecem soluções para a tarefa de clusterização. Esse método consegue ser mais preciso do que algoritmos de clusterização individuais, pois, melhora o resultado compensando erros cometidos por algum algoritmo pela intervenção da solução correta de outros.

Para combinar os clusters gerados pelos algoritmos é utilizada uma função de consenso. O método de ensemble de clusters permite a geração de clusters de melhor qualidade e a obtenção de configurações inalcançáveis por um único algoritmo. Além disso, os resultados gerados são menos susceptíveis a ruídos, variações amostrais e são capazes de integrar soluções baseadas em múltiplas fontes de dados ou atributos distribuídos (NGUYEN; CARUANA, 2007).

O Particle Swarm Optimization (PSO) é um algoritmo estocástico de base populacional, proposto por Kennedy e Eberhart (1995), inspirado no comportamento social de animais como cardume de peixes e revoada de aves. Com um esquema de busca estocástica, o PSO tem características de computação simples e capacidade de rápida convergência. O PSO foi aplicado com sucesso em diversas áreas, tais como problemas de clusterização (ALAM; DOBBIE; RIDDLE, 2008; ESMIN; Coelho; Matwin, 2013), e processamento de imagem (NIU; SHEN, 2006; OMRAN; SALMAN; ENGELBRECHT, 2006). Além disso, o PSO provou ser competitivo frente a algoritmos genéticos em várias tarefas, principalmente, na área de otimização (ESMIN; LAMBERT-TORRES, 2012; SILVA; NEVES; COSTA, 2002; ZHAO; GUO; CAO, 2005).

1.2 Objetivo

Esse trabalho foi realizado com o objetivo de propor a utilização do algoritmo PSO para atuar nas duas fases de um *ensemble* de *clusters*. Investigar a eficácia do algoritmo PSO para esse problema, independente de como os dados são estruturados e passados para a função de consenso. Por um estudo empírico compara-se a precisão do método proposto com outros métodos de clusterização usados como função de consenso.

1.3 Objetivos Específicos

Para atender ao objetivo, é necessário que os seguintes objetivos específicos sejam atendidos:

a) compreender o algoritmo *Particle Swarm Optimization* para clusterização de dados;
b) compreender o método de *ensemble* de *clusters*;
c) estudar os diferentes métodos de função de consenso presentes na literatura.
d) implementar diferentes medidas de similaridade para clusterização;
e) utilizar o algoritmo *Particle Swarm Optimization;*
f) investigar o uso do *ensemble* de *cluster;*
g) realizar experimentos;
h) análise dos resultados dos experimentos.

1.4 Estrutura do trabalho

O presente trabalho está dividido em 5 capítulos. O capítulo 1 faz uma breve introdução, seguida de uma contextualização e dos objetivos. O referencial

teórico é apresentado no Capítulo 2. A metodologia utilizada neste trabalho é apresentada no Capítulo 3. O Capítulo 4 contém os experimentos realizados, junto com os resultados e discussões. As conclusões resultantes deste trabalho estão no capítulo 5.

2 REFERENCIAL TEÓRICO

Neste capítulo são introduzidos os conceitos de mineração e clusterização de dados para o melhor entendimento do trabalho. São apresentadas as medidas de similaridades, o algoritmo *Particle Swarm Optimization* e sua modificação para a tarefa de clusterização bem como o conceito de *ensemble* de *clusters*.

2.1 Mineração de dados

Nas últimas décadas tem acontecido um crescimento na aquisição e armazenamento de dados, e esses dados, por muitas vezes, possuem uma gama de conhecimento oculto muito importante. A tarefa de extrair conhecimento, com base nesses dados armazenados, tem se tornado cada vez mais difícil e complicada, superando a capacidade humana. Para enfrentar esse desafio, surgiu uma área chamada *Data Mining* (DM) ou mineração de dados (GHOSH; JAIN, 2005). A análise e extração de conhecimento, considerando uma base de dados é um processo de busca e identificação de novos padrões, que consiste de várias técnicas, entre elas, a mineração de dados.

A mineração de dados é definida como o processo de descoberta de padrões em banco de dados. O processo deve ser automático ou mais usualmente semiautomático. O processo de mineração de dados consiste em extrair padrões úteis ou minerar conhecimento de grandes quantidades de dados por meio da aplicação de algoritmos apropriados, ferramentas e técnicas. Os padrões descobertos devem ser significativos na medida em que leve a alguma vantagem, geralmente de caráter econômico (WITTEN; TIBSHIRANI, 2010).

Segundo Han e Kamber (2005), *Data Mining* é uma área multidisciplinar usada para extrair conhecimento, com base em grandes volumes de dados, utilizando conhecimento de áreas como *Machine Learning*, estatística entre outras. O processo de *Knowledge Discovery in Databases* (KDD) consiste de várias etapas nas quais a mineração de dados está presente como pode ser observado na Figura 1.

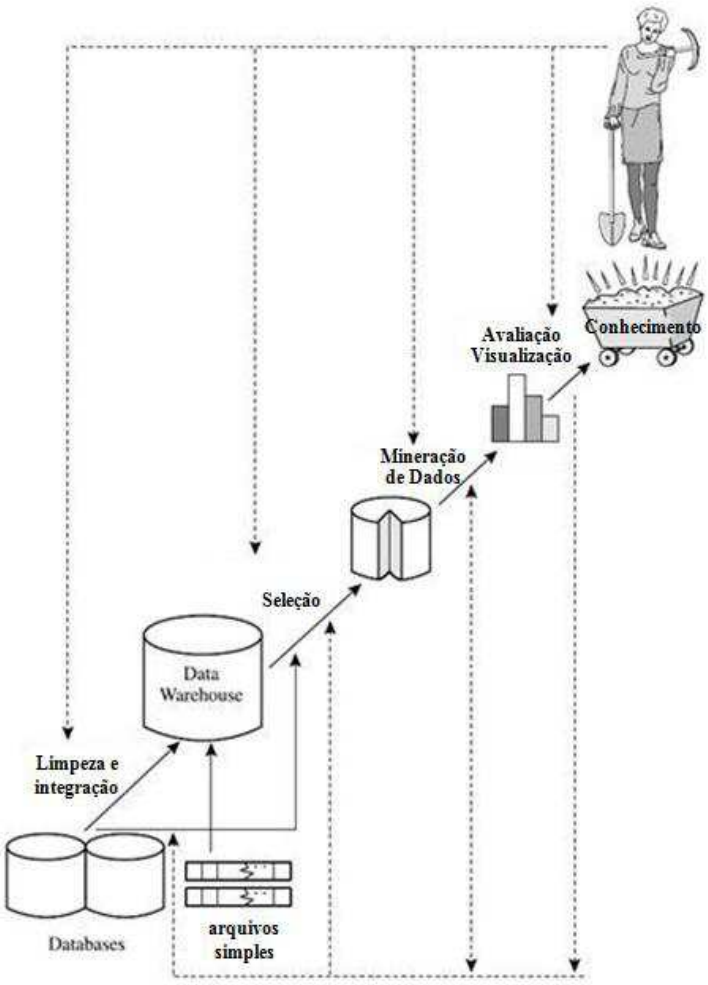

Figura 1 *Knowledge Discovery in Databases* (KDD), imagem adaptada de Han e Kamber (2005)

Ghosh e Jain (2005) descrevem a mineração de dados como uma área de *Knowledge Discovery in Databases* (KDD), sendo ela uma parte deste processo. O KDD é o nome que se dá ao processo global de descoberta de informações úteis nas bases de dados. Segundo eles, o processo de KDD consiste em três etapas; pré-processamento ou preparação de dados, mineração de dados ou aplicação de algoritmos e pós-processamento ou interpretação do conhecimento. Após a terceira etapa ser completada serão avaliados os padrões apresentados.

2.2 Clusterização

Clustering é um conjunto de técnicas usadas no processo de particionamento de um conjunto de dados não rotulados em grupos de objetos semelhantes. Os dados são arranjados em grupos, chamados de *cluster*, que consiste de objetos semelhantes entre si no que diz respeito à certa medida de similaridade e que são diferentes de objetos de outros grupos (COLE, 1998). A Figura 2 exemplifica o processo de clusterização.

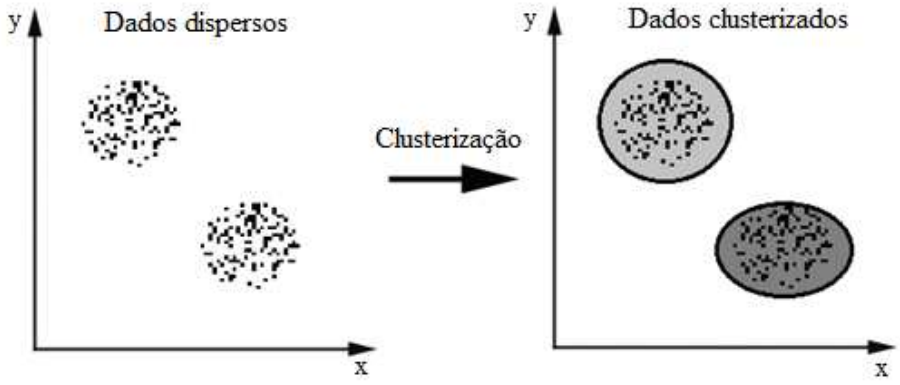

Figura 2 Processo de Clusterização

A clusterização é um processo que antecede a classificação, já que os dados nesse momento não possuem rótulo. Após clusterizados, os dados em cada *cluster* poderão ser analisados e identificados com base em características que fizeram com que estivessem naquele *cluster*. Dessa forma, é possível classificar os elementos daquele grupo com base na característica em comum do grupo (KRIEGEL; KRÖGER; ZIMEK, 2009; LUXBURG, 2007).

O processo de clusterização de dados pode ser definido da seguinte forma: seja um conjunto de dados de *n* objetos $X = \{X_1 , X_2 , ..., X_n \}$, onde cada

$X_i \in \mathbb{R}^p$ é um vetor de *p* medidas reais, que serão clusterizados

$C = \{C_1 , C_2 , ... ,C_k \}$, onde *k* é o número de *clusters*, de forma que respeite 3 condições básicas:

a) $C_i \neq \{ \}$, não deve existir *cluster* vazio;

b) $C_i \cap C_j = \{ \}$, dois *clusters* diferentes não devem ter nenhum objeto em comum;

c) $C_1 \cup C_2 ... \cup C_k = X$, cada objeto deve definitivamente ser anexado a um *cluster*, de forma que nenhum objeto esteja sem *cluster*.

Segundo Han e Kamber (2005), existem cinco principais métodos de clusterização, que são:

a) Métodos Hierárquicos: em que se cria uma decomposição hierárquica de um dado conjunto de dados e cuja forma mais comum de representação é feita atrás de um dendrograma. Na base se encontra cada objeto da base de dados e a média em que se sobe um nível, os ramos do dendrograma vão se unindo em grupos de objetos semelhantes, até que no topo exista apenas um ramo.

b) Métodos Baseados em Grid: nesses métodos o espaço onde os dados estão dispostos são divididos em uma estrutura finita de grades (grid). A operação de clusterização é executada sobre a estrutura de grade, o que garante ganho de velocidade no processamento, porque independe da quantidade de dados que está dentro das grades, mas, sim, da quantidade de grades existentes naquele espaço.

c) Métodos Baseados em Modelos: os métodos de clusterização, baseados em modelos, são geralmente robustos e pouco sensíveis a ruídos. Esse método trabalha hipotetizando modelos, para cada um dos clusters a ser criado, de forma que os dados ali presentes se encaixem da melhor forma possível ao modelo.

d) Métodos Baseados em Densidade: métodos desenvolvidos para clusterização de dados dispersos no espaço de busca de forma irregular. Esse método cria clusters em regiões no espaço onde a distribuição dos objetos é mais densa, e os clusters são separados por regiões no espaço em que a densidade dos objetos é muito baixa ou nula.

e)Métodos de Particionamento: nesses métodos uma base de dados com n objetos é dividida em k partições, de forma que, a cada iteração, os objetos são realocados na partição que melhor se ajustem. Os dados são reajustados de forma que os objetos semelhantes entre si estejam presentes em uma mesma partição e que se diferem de objetos que estejam presentes em outra partição.

O real valor do processo de clusterização é, geralmente, medido de forma subjetiva, em termos do quão útil os resultados se apresentam.

2.3 Medidas de similaridade

Medida similaridade ou distância entre dois pontos de dados é um requisito fundamental para várias tarefas de mineração de dados e descoberta de conhecimento. Na formação dos *clusters*, a pequena distância entre dois objetos X_i e X_j significa grande similaridade, ou seja, elementos que estarão em um mesmo *cluster*. Também pode ser usada para quantificar a dissimilaridade, o que determina que um elemento esteja em outro *cluster* (COLE, 1998; WOLFRAM, 2013). O termo *t* se refere ao *t*-ésimo atributo dos objetos X_{it} e X_{jt} . As medidas de similaridade que serão utilizadas neste trabalho são:

- Distância Euclidiana.

$$d(X_i, X_j) = \sqrt{\sum_{p=1}^{t} (X_{ip} - X_{jp})}$$

- Distância *Manhattan* ou *City Block.*

$$d(X_i, X_j) = \sum_{p=1}^{t} |X_{ip} - X_{jp}|$$

- Distância *Chessboard.*

$$d(X_i, X_j) = max_{c=1,\ldots,t}\{|X_{ic} - X_{jc}|\}$$

- Distância Cosseno.

$$d(X_i, X_j) = 1 - \frac{\sum_{p=1}^{t}(X_{ip} \times X_{jp})}{\sqrt{\sum_{p=1}^{t}|X_{ip}|^2} \times \sqrt{\sum_{p=1}^{t}|X_{jp}|^2}}$$

- Distância Canberra.

$$d(X_i, X_j) = \sum_{p=1}^{t} \frac{(|X_{ip} - X_{jp}|)}{|X_{ip}| + |X_{jp}|}$$

Antes da clusterização, uma medida de similaridade deve ser determinada. A medida reflete o grau de proximidade ou de separação dos objetos e deve corresponder às características que são usadas para distinguir os *clusters* nos quais os dados são atribuídos. Em muitos casos, estas características são dependentes dos dados ou do contexto do problema, e não há nenhuma medida que é universalmente a melhor para todos os tipos de problemas de clusterização.

2.4 *Particle Swarm Optimization*

O algoritmo *Particle Swarm Optimization* (PSO) é um método de otimização baseado na simulação do comportamento social de bandos de pássaros. Geralmente enquadrado na computação evolutiva, o PSO tenta

encontrar a solução ideal utilizando uma população de partículas (KENNEDY; EBERHART, 1995; SHI; EBERHART, 1998).

O PSO mantém um enxame de partículas, em que cada partícula representa um potencial candidato para a solução do problema. As partículas cooperam entre si para encontrar a melhor posição (melhor solução) no espaço de busca (espaço de solução).

Cada partícula move de acordo com a sua velocidade. Em cada iteração, o movimento das partículas é calculado pelas equações de posição e velocidade como se segue:

$$x_i\ (t\ +1) = x_i\ (t) + v_i\ (t)$$
$$v_i\ (t\ +1) = \omega v_i(t) + c_1 r_1 (pbest_i(t) - x_i(t)) + c_2 r_2 (gbest(t) - x_i(t))$$

Nas equações de posição e velocidade, $x_i(t)$ é a posição de partícula *i* no instante *t*, $v_i(t)$ é a velocidade da partícula *i* no instante *t*, $pbest_i(t)$ é a melhor posição encontrada pela própria partícula até então, $gbest(t)$ é a melhor posição encontrada pelo enxame até então, ω é o valor do peso inercial, c_1 e c_2 são dois coeficientes que regulam o passo máximo na direção da melhor posição pessoal ($pbest_i(t)$) e melhor posição global ($gbest(t)$) da partícula, e r_1 e r_2 são variáveis aleatórias dentro do intervalo de 0 e 1 que contribuem para a natureza estocástica do algoritmo (ALAM; DOBBIE; RIDDLE, 2008; YANG; KAMEL, 2003).

As equações $pbest_i(t)$ e $gbest(t)$ definem como os melhores valores pessoal e global são atualizados no tempo *t*, respectivamente. Suponha que o enxame consiste de *s* partículas. Assim, $i \in 1...s$

$$pbest_i(t+1) = \begin{cases} pbest_i(t) & se \quad f(pbest_i(t)) > x_i(t+1)) \\ x_i(t) & se \quad f(pbest_i(t)) \leq x_i(t+1)) \end{cases}$$

$$gbest(t+1) = \min\{f(y), f(gbest(t))\}, \text{ em que}$$

$$y \in \{pbest_0, pbest_1, \ldots, pbest_s\}$$

O esquema de atualização da posição de uma partícula é mostrado na Figura 3.

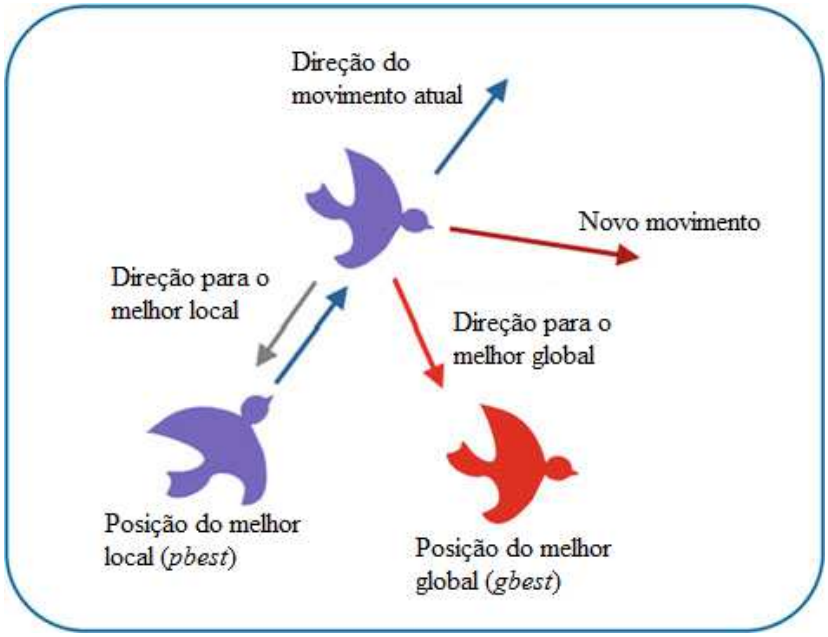

Figura 3 Atualização da posição de uma partícula, figura adaptada de Ahmadi, Karray e Kamel (2010)

O processo do PSO é mostrado na Figura 4.

```
Inicializar uma população de partículas i.
Enquanto(as condições de término não são satisfeitas){
    Para cada partícula i faça{
         Atualizar i de acordo com as equações de
         posição e velocidade.

         Calcular o fitness.

         Atualizar o melhor valor pessoal e melhor valor
         global de acordo com as equações de pbest_i(t) e
         gbest(t).
    }
}
```

Figura 4 Algoritmo PSO

A descrição do funcionamento do processo mostrado na Figura 4: inicialmente, com base em informações do *fitness* de cada partícula, uma partícula é identificada como a partícula com a melhor posição global. A função de *fitness* é usada para avaliar a posição da partícula no espaço de busca, ou seja, o *fitness* de cada partícula significa quão boa ela é para resolver o problema. Em seguida, todas as partículas serão aceleradas na direção a essa partícula, mas, ao mesmo tempo, na direção dos seus próprios melhores locais previamente calculados. Ocasionalmente, as partículas irão ultrapassar o alvo, explorando o espaço de busca além dos atuais valores de melhor local da partícula. Todas as partículas, também, têm a chance de descobrir o melhor global, caso em que as outras partículas vão mudar de direção e seguir em direção da partícula com o novo valor de melhor global. Ao abordar o espaço, onde a partícula com o melhor global está por diferentes direções no espaço de busca, as chances de que essas soluções vizinhas sejam descobertas por alguma outra partículas são boas.

O algoritmo PSO é muito rápido, simples e fácil de compreender e aplicar. Também há poucos parâmetros para ajustar (KENNEDY; KENNEDY; EBERHART, 2001). O PSO encontra o melhor valor com a interação entre partículas, mas, quando o espaço de busca é alto, a sua velocidade de

convergência torna-se muito lenta, próximo do ótimo global. Também apresenta resultados ruins quando lida com um conjunto de dados grandes e complexos.

2.5 PSO para clusterização

A utilização do PSO para clusterização é apresentada em Merwe e Engelbrecht (2003), em que cada partícula representa uma possível solução do problema. Segundo este modelo, cada partícula é representada por um vetor de tamanho N_c de centroides, em que N_c é o número máximo de *clusters* que podem ser criados. A partícula x_i é construída da seguinte forma:

$$x_i = (m_{i,1}, ..., m_{i,j}, ..., m_{i,Nc})$$

em que $m_{i,j}$ corresponde ao *j*-ésimo centroide da *i*-ésima partícula em um *cluster* $C_{i,j}$. Com essa estrutura, cada partícula representa uma solução candidata no *swarm*. A função de *fitness* das partículas é facilmente medida como a quantização do erro, da seguinte forma:

$$f = \frac{\sum_{j=1}^{N_c} \left[\sum_{\forall Z_p \in C_{i,j}} \frac{d(Z_p, m_{i,j})}{|C_{i,j}|} \right]}{N_c}$$

na qual *i* é a partícula, Z_p denota o *p*-ésimo elemento no vetor de dados do *cluster* $C_{i,j}$ e *d* é a medida de similaridade entre Z_p e $m_{i,j}$. $|C_{i,j}|$ é a quantidade de dados pertencentes ao *cluster* $C_{i,j}$. O algoritmo PSO para clusterização é apresentado na Figura 5.

```
Inicializar uma população de partículas i.
Enquanto(as condições de término não são verdadeiras){
    Para cada partícula i faça{
        Para cada dado Zp{
            Calcular d(Zp, mi,j) para todos os centroides.
            Atribuir Zp ao cluster Ci,j, tal que

            d(Zp, mi,j) = min∀c=1...Nc {d(Zp, mi,c)}

        }
        Calcular o fitness da partícula de acordo com a
        equação f .

        Atualizar os valores de melhor local e melhor
        global de acordo com as equações de pbesti(t) e
        gbest(t).
```

Figura 5 Algoritmo PSO para clusterização

2.6 *Cluster Ensemble*

A clusterização, muitas vezes, é o primeiro passo na análise de dados. Existem diversos métodos de clusterização já desenvolvidos, tais como métodos hierárquicos, métodos baseados em densidade, em particionamento, em *grid*, entre outros. A maioria dos métodos de clusterização se concentra em encontrar *clusters* ideais ou próximos do ideal, de acordo com algum critério de clusterização específico. O *ensemble* de *clusters* pode fornecer benefícios além do que um algoritmo de agrupamento único pode alcançar. O *ensemble* de *clusters*, muitas vezes, gera melhores *clusters*, encontra um *cluster* combinado inatingível por qualquer algoritmo único de clusterização, é menos sensível a ruídos, *outliers* ou variações da amostra (NGUYEN; CARUANA, 2007).

O método de *ensemble* de *clusters* utiliza-se de vários resultados de diferentes algoritmos de clusterização em uma solução de consenso para buscar a melhoria da qualidade e solidez dos resultados. O *ensemble* de *clusters*

trabalha compensando a possibilidade de erros cometidos por alguns algoritmos de clusterização pela intervenção da solução correta de outros, sendo este modelo mais preciso do que algoritmos de *clusters* individuais (GHAEMI et al., 2009; STREHL; GHOSH, 2003).

O *ensemble* de *clusters* é composto de duas fases. Na primeira fase, o conjunto de algoritmos que compõe o *ensemble* recebe a base de dados e tem como saída um conjunto de *clusters* como solução. A segunda fase recebe o conjunto de *clusters* como entrada e os combina por meio de uma função de consenso produzindo *clusters* como saída final (FERN; BRODLEY, 2004; GHAEMI et al., 2009). As duas fases do *ensemble* são descritas, formalmente, a seguir:

***Ensemble*:** Dado um conjunto de dados com n instâncias $X = \{x_1,x_2,\ldots,x_n\}$ e um conjunto formado por r algoritmos de clusterização $E = \{a_1,a_2,\ldots,a_r\}$, cada solução de a_i é representada por um conjunto disjunto de *clusters* representado por $\lambda_i = \{c_1,c_2,\ldots,c_k\}$, em que $\cup_k \; c_k^i = X$, no qual k é o número de *clusters* produzido por cada algoritmo do *ensemble*.

Consenso: Dado um *ensemble* de *clusters* E e o número k, uma função de consenso Γ usa a informação provida de E para gerar k *clusters* finais. Em alguns casos, as informações contidas em X são usadas para gerar os *clusters* finais.

A Figura 6 ilustra um *ensemble* de *clusters*, a função de consenso Γ combina os resultados λ_i de um conjunto de algoritmos de clusteriação E, produzindo um resultado λ final.

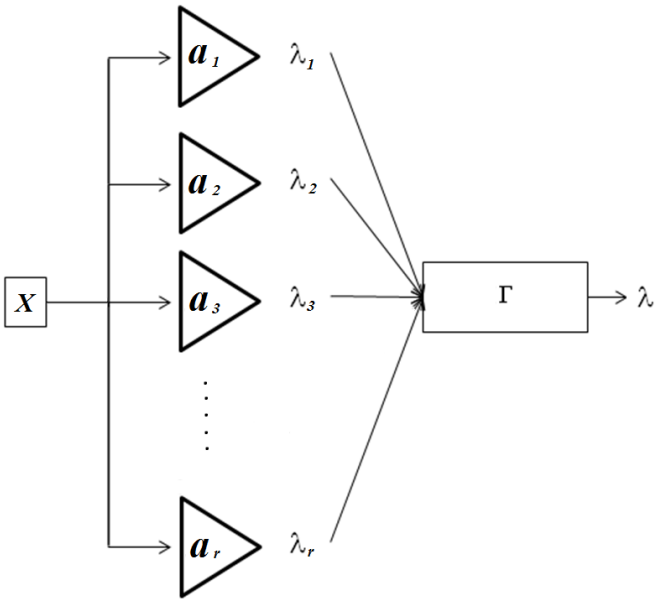

Figura 6 Arquitetura de um *ensemble* de *clusters*

Na literatura existem alguns tipos de função de consenso, como:

a) Hypergraph Consensus Methods: em que os clusters podem ser representados como hyperedges em um grafo cujos vértices correspondem aos objetos a serem clusterizados de modo que cada hyperedge descreve um conjunto de objetos pertencentes aos mesmos clusters. O problema da função de consenso é, então, reduzido a encontrar o corte mínimo de um hipergrafo. O corte mínimo deste hipergrafo em k componente irá fornecer o consenso dos clusters (NGUYEN; CARUANA, 2007; TOPCHY; JAIN; PUNCH, 2004).

b) Mutual Information Approach: nessa abordagem, a função objetivo para um ensemble de clusters pode ser formulada como uma informação mútua entre a distribuição de probabilidade empírica dos

rótulos na partição consenso e os rótulos no ensemble (NGUYEN; CARUANA, 2007; TOPCHY; JAIN; PUNCH, 2004).

c) Co-association based functions: nessa abordagem, a semelhança entre as instâncias pode ser estimada pelo número de clusters compartilhados em que as instâncias estejam presentes nos clusters do ensemble. A função de consenso opera em uma matriz de coassociação em que vários algoritmos hierárquicos algomerativos podem ser aplicados para obter os clusters finais. Os valores da matriz de coassociação são usados na função de fitness (NGUYEN; CARUANA, 2007; TOPCHY; JAIN; PUNCH, 2005; YANG; ZHANG; WANG, 2009).

d) Mixture Model Approach: Nesse modelo os rótulos são modelados como variáveis aleatórias tiradas de uma distribuição de probabilidade descrita como uma mistura de densidades de componentes multinominais. O objetivo da clusterização por consenso é formulado como um problema para estimar a máxima verossimilhança. Para encontrar a densidade de mistura adequada para um certo dado, é preciso maximizar a função de verossimilhança no que diz respeito aos parâmetros não conhecidos (GHAEMI et al., 2009; TOPCHY; JAIN; PUNCH, 2004).

Re-labeling Approach ou *Voting Approach*: o método por votação procura resolver o problema do consenso por meio do simples processo de correspondência no qual associa os objetos a determinados *clusters* pela contagem majoritária de seus rótulos. No entanto, a correspondência entre rótulos é que faz esse problema difícil. Para isso, podem-se permutar os rótulos dos *clusters* de forma que o melhor ajuste entre os rótulos e *clusters* seja obtido. Uma técnica é fazer a re-rotulação de todas as partições do conjunto de acordo

com uma partição de referência (NGUYEN; CARUANA, 2007; TOPCHY; JAIN; PUNCH, 2004).

2.7 Trabalhos relacionados

Nessa seção é apresentado um breve resumo dos trabalhos que mais se relacionam ao trabalho proposto.

No trabalho do Yang e Kamel (2013) foi apresentado o ensemble de clusters, usando três colônias de formigas, em que cada colônia utiliza de um modelo de função para a velocidade da formiga diferente: constante, randômico e randomicamente decrescente. Essa abordagem se baseia no modelo hypergraph para combinar os clusters produzidos pelos três diferentes algoritmos de clusterização. Os resultados experimentais mostraram que o ensemble melhorou a qualidade dos clusters. No trabalho realizado por Nguyen e Caruana (2007) foram apresentados três métodos, baseados no algoritmo EM, para atuar como função de consenso. Um estudo empírico comparou os métodos propostos com outros onze métodos de função de consenso, em quatro bases de dados, usando seis diferentes métricas de qualidade de cluster. Os experimentos mostraram que o método proposto conseguiu construir clusters tão bons quanto os outros métodos.

Topchy, Jain e Punch (2004) apresentaram um modelo probabilístico de função de consenso utilizando uma mistura finita de distribuições multinomiais em um espaço de clusters. Foi feito um estudo comparando o desempenho do método proposto outros baseados no algoritmo EM. Dentre as vantagens do método proposto, está a baixa complexidade computacional e o fato de ser um modelo estatístico bem fundamentado. Pelos resultados experimentais deduz-se a eficácia do método proposto em bases de dados do mundo real. Nisha, Mohanavalli e Swathika (2013) apresentaram dois métodos para melhorar a

precisão e qualidade do processo de clusterização. O primeiro método foi o método de coassociação em que a comparação do par sábio é feito e o fator peso decide o rótulo do dado. O segundo é o método da informação mútua normalizada, em que a informação compartilhada entre dois clusters são medidos e agregado aos clusters. Nos experimentos foram observados que os resultados gerados pelo ensemble de clusters são mais robustos e precisos.

Yang, Zhang e Wang (2009) propuseram um novo modelo de combinação ponderada de múltiplas partições, no qual o algoritmo de otimização por enxame de partículas foi utilizado para otimizar o parâmetro. Os resultados experimentais indicam que a abordagem pode gerar cluster de melhor qualidade em comparação com um algoritmo único de clusterização. No entanto, o método proposto não funciona bem com bases de dados pequenos em virtude de problemas de sobremontagem. No trabalho de Kuncheva, Hadjitodorov e Todorova (2006) foi feita a comparação experimental de 24 métodos de ensemble de clusters. Foram usadas 24 bases de dados, tanto reais quanto artificiais. Para avaliar foram usados o índice Rand ajustado e a precisão na classificação, tendo como critério o conhecimento do atributo de classe dos dados. Funções de consenso que interpretavam a matriz consenso, proveniente do conjunto dados, em vez de usar funções de similaridade, tiveram melhores resultados que as alternativas tradicionais, incluindo CSPA e HGPA.

O artigo de Fern e Brodley (2004) apresentou o método HBGF, no qual formula um grafo que reduz o problema do ensemble de cluster a um problema de particionamento de grafo bipartido. Pelos experimentos constata-se que o HBGF alcança desempenho comparável, ou melhor, em comparação com as outras duas abordagens. Li e Ding (2008) propuseram um framework para clusterização por consenso. Nesse framework, cada entrada dada é ponderada, e os pesos determinam a média que produz resultados de melhor qualidade. Foi

mostrada que a clusterização por consenso ponderado resolve o problema de redundância em que existe uma alta correlação nos clusters de entrada.

3 O MÉTODO PROPOSTO

Objetivou-se neste trabalho investigar o desempenho do algoritmo *Particle Swarm Optimization* atuando tanto como algoritmo de clusterização quanto como função de consenso no *ensemble* de *clusters*. Outros algoritmos de clusterização serão utilizados para fins de comparação e diversificação do *ensemble.*

Para atender a esse objetivo, será apresentado o modelo do *ensemble* utilizado, além de dois tipos de base dados que será usada como entrada para a função de consenso. A forma como cada experimento foi realizado está descrita no capítulo 4.

3.1 *Ensemble* proposto

O *ensemble* proposto é dividido em duas fases: a primeira com um conjunto de algoritmos de clusterização, gerando *clusters* com base em uma mesma base de dados. A segunda, com a re-rotulação dos *clusters* criados, considerando uma partição de referência escolhida *a priori* e a aplicação da função de consenso. A função de consenso utiliza o *Re-labeling Approach* somente para alcançar a correspondência dos rótulos dos diversos *clusters* criados pelos algoritmos de clusterização.

A Figura 7 ilustra o *ensemble* de *clusters*, uma mesma base de dados X é dada como entrada para r algoritmos de clusterização a_1, a_2, ... a_r. Os *clusters* gerados por cada algoritmo de clusterização λ_1, λ_2, ..., λ_r serão re-rotulados de acordo com um uma partição de referência λ_{ref} . Uma função de consenso Γ combina os resultados dos r algoritmos de clusteriação, produzindo o resultado λ final.

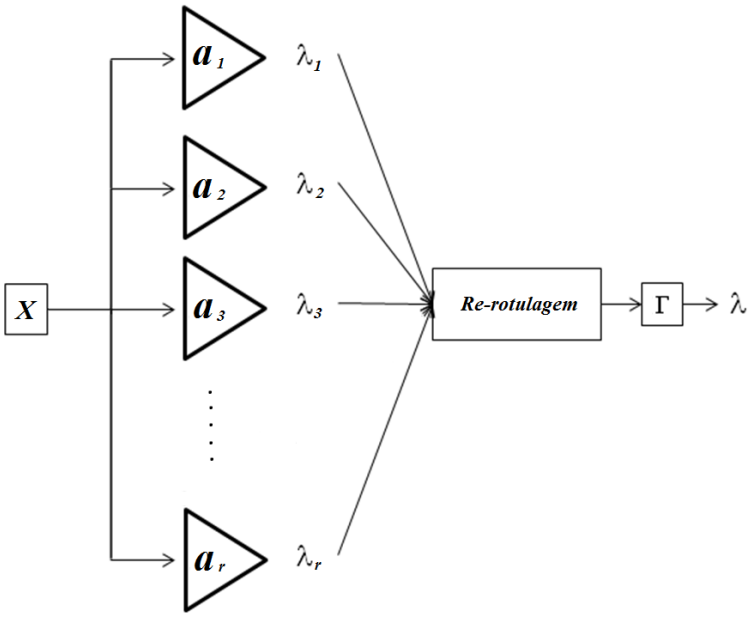

Figura 7 Modelo *Ensemble* de *clusters* utilizado no trabalho

3.2 Rotulação

Ao se criar *clusters*, considerando uma mesma base de dados, utilizando dois ou mais algoritmos de clusterização diferentes, é esperado que tanto os *clusters* criados quanto os rótulos empregados a esses *clusters* não sejam idênticos. A Figura 8 exemplifica a clusterização de uma mesma base de dados feita por dois diferentes algoritmos de clusterização, está disposta a distribuição espacial dos dados de um dos atributos da base de dados *Wine* (ASUNCION; NEWMAN, 2007) clusterizado pelos algoritmos *K-means* e *Expectation-maximization* (EM) usando a ferramenta Weka (HALL et al., 2009).

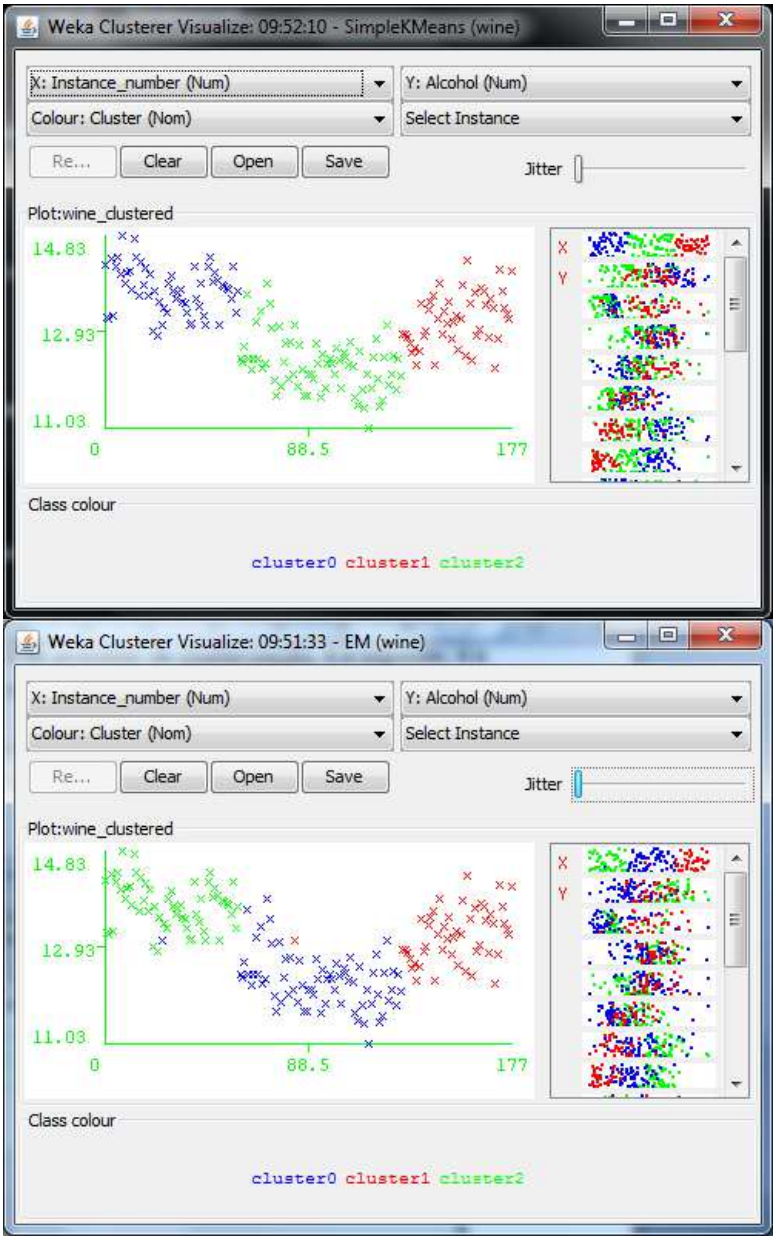

Figura 8 Resultado da clusterização dos algoritmos K-means e EM, respectivamente, considerando o Weka

Formalizando o problema da rotulação, temos um dado conjunto formado por *r* algoritmos de clusterização $E = \{a_1,a_2,\ldots,a_r\}$, cada solução de a_i é representada por um conjunto disjunto de *clusters*, representado por $\lambda_i = \{c_1,c_2,\ldots,c_k\}$, cada conjunto λ_i poderá rotular os *clusters*, criados de forma diferente, sendo necessária uma nova rotulação de acordo com uma λ_{ref} de referência. A Tabela 1 exemplifica uma base de dados com 6 instâncias, um *ensemble* com quatro algoritmos e seus resultados rotulados de forma diferente.

Tabela 1 Exemplo de uma base de dados com 6 componentes, um *ensemble* com 4 algoritmos e seus resultados rotulados de forma diferente

Instâncias	a_1	a_2	a_3	a_4 *(Referência)*
x_1	0	1	1	0
x_2	0	1	1	0
x_3	1	1	0	0
x_4	1	0	0	1
x_5	1	0	0	1
x_6	0	0	1	1

Observando a solução apresentada pelos algoritmos a_2 e a_4, percebe-se que elas são idênticas, mudando apenas a rotulação. O mesmo acontece com as soluções apresentadas pelos algoritmos a_1 e a_3. É necessário, então, que seja feita uma ré-rotulagem dos λ_i para que a função de consenso possa ser executada. Dado que a função de consenso conheça o valor de *k*, para este trabalho foi feita uma ré-rotulagem de todo λ_i com o seu melhor acordo com uma λ_{ref} de referência escolhida *a priori* (TOPCHY; JAIN; PUNCH, 2004). A Tabela 2 apresenta os resultados da Tabela 1 após a re-rotulagem.

Tabela 2 Resultado da re-rotulagem sobre os dados da Tabela 1

Instâncias	a_1	a_2	a_3	a_4 ***(Referência)***
x_1	0	0	0	0
x_2	0	0	0	0
x_3	1	0	1	0
x_4	1	1	1	1
x_5	1	1	1	1
x_6	0	1	0	1

3.3 Construção das bases de dados

Para este trabalho foram construídas duas diferentes estruturas de dados que irão ser utilizadas pela função de consenso. A primeira estrutura é formada somente pelos rótulos dos dados clusterizados, e a segunda formada pelos dados utilizados na clusterização acrescidos dos rótulos.

Na estrutura formada apenas pelos rótulos, cada algoritmo de clusterização é transformado em um atributo, ou seja, dado um *ensemble* com r algoritmos de clusterização, a nova base de dados formada apenas pelos rótulos terá $r + 1$ atributos, o atributo a mais é o atributo de classe. A Figura 9 mostra parte da base de dados *Iris* (ASUNCION; NEWMAN, 2007), que possui cinco atributos, sendo um atributo de classe.

```
@relation iris

@attribute sepallength  REAL
@attribute sepalwidth   REAL
@attribute petallength  REAL
@attribute petalwidth   REAL
@attribute class    {Iris-setosa,
                     Iris-versicolor,
                     Iris-virginica}

@data
5.1,3.5,1.4,0.2,Iris-setosa
4.9,3.0,1.4,0.2,Iris-setosa
4.7,3.2,1.3,0.2,Iris-setosa
4.6,3.1,1.5,0.2,Iris-setosa
5.0,3.6,1.4,0.2,Iris-setosa
5.4,3.9,1.7,0.4,Iris-setosa
```

Figura 9 Parte da base de dados *Iris*

A Figura 10 mostra parte da base de dados, formada com apenas rótulos, considerando a base de dados *Iris* da Figura 9. Os atributos dessa base de dados é constituído por seis algoritmos de clusterização, que são; o PSO para clusterização, com cinco diferentes medidas de similaridade: distância Euclidiana (PSOEuc), distância *Manhattan* (PSOMan), Distância Cosseno (PSOCos), Distância *Chessboard* (PSOChes) e Distância Canberra (PSOCan) (COHEN; CASTRO, 2006; ESMIN; PEREIRA; ARAUJO, 2008; MERWE; ENGELBRECHT, 2003; WOLFRAM, 2013), bem como o *K-means* com a função de similaridade euclidiana (KMEuc) (VAIDYA; CLIFTON, 2003). Cada instância é formada pelos rótulos resultantes do processo de clusteriazão de cada algoritmo.

```
@relation iris

@attribute KMEuc        REAL
@attribute PSOEuc       REAL
@attribute PSOMan       REAL
@attribute PSOCos       REAL
@attribute PSOChes      REAL
@attribute PSOCan       REAL
@attribute class    {0, 1, 2}

@data
0,0,0,0,0,0,0
0,0,0,0,0,0,0
0,0,0,0,0,0,0
0,0,0,0,0,0,0
0,0,0,0,0,0,0
0,0,0,0,0,0,0
0,0,0,0,0,0,0
0,0,0,0,0,0,0
0,0,0,0,0,0,0
0,0,0,0,0,0,0
0,0,0,0,0,0,0
0,0,0,0,0,0,0
```

Figura 10 Rótulos criados considerando a base de dados *Iris*

Na estrutura formada pelos dados acrescidos dos rótulos, cada algoritmo de clusterização terá os rótulos de seus *clusters* transformados em um atributo e acrescido a base de dados, ou seja, dado uma base de dados com *d* atributos, um *ensemble* com *r* algoritmos de clusterização, a nova base de dados formada pelos dados e pelos rótulos terá $d + r + 1$ atributos, o atributo a mais é o atributo de classe.

A Figura 11 mostra parte da base de dados formada com os dados acrescidos dos rótulos considerando a base de dados *Iris* clusterizada por seis algoritmos.

```
@relation iris

@attribute sepallength  REAL
@attribute sepalwidth   REAL
@attribute petallength  REAL
@attribute petalwidth   REAL
@attribute KMEuc REAL
@attribute PSOEuc REAL
@attribute PSOMan REAL
@attribute PSOCos REAL
@attribute PSOChes REAL
@attribute PSOCan REAL
@attribute class    {0, 1, 2}

@data
5.1,3.5,1.4,0.2,0,0,0,0,0,0,0
4.9,3.0,1.4,0.2,0,0,0,0,0,0,0
4.7,3.2,1.3,0.2,0,0,0,0,0,0,0
4.6,3.1,1.5,0.2,0,0,0,0,0,0,0
5.0,3.6,1.4,0.2,0,0,0,0,0,0,0
5.4,3.9,1.7,0.4,0,0,0,0,0,0,0
4.6,3.4,1.4,0.3,0,0,0,0,0,0,0
5.0,3.4,1.5,0.2,0,0,0,0,0,0,0
```

Figura 11 Dados acrescidos de rótulos criados considerando a base de dados *Iris*

Será investigada a eficácia do algoritmo PSO para esse problema, independente de como os dados são estruturados e passados para a função de consenso.

4 RESULTADOS E DISCUSSÃO

Nessa seção são apresentados e discutidos os principais resultados da pesquisa. Os resultados estarão dispostos em experimentos que foram feitos durante todo o decorrer do trabalho, e alguns deles já publicados em Esmin e Coelho (2013) e Esmin, Coelho e Matwin (2013) e o artigo de Coelho e Esmin (2013) fruto do Experimento 1. Para todos os experimentos, primeiramente estão descritos os algoritmos usados para criar o *ensemble* e, em seguida, as funções de consenso utilizadas.

4.1 Experimento 1 - *Ensemble* de *cluster* usando PSO

Nesse experimento, o algoritmo PSO foi empregado nas duas fases do *ensemble* de *clusters*, a fim de verificar o comportamento do PSO frente a outros algoritmos. Na primeira fase, o PSO participa do *ensemble* junto com outros quatro algoritmos de clusterização e o algoritmo com maior taxa de erro não será utilizado como função de consenso. Na segunda fase o PSO foi usado como função de consenso e seu resultado comparado com o dos outros três algoritmos.

4.1.1 Base de dados

Os experimentos foram feitos usando bases de dados artificiais e reais. Foram usadas seis bases de dados, três delas reais (*Iris*, *Diabetes* e *Yeast*), obtidas do repositório de referência da UCI (ASUNCION; NEWMAN, 2007) e três bases de dados sintéticos, *two-spiral* (JIA; CHUA, 1995), *spiral* (CHANG; YEUNG, 2008) e *half-rings* (JAIN; LAW, 2005). A Figura 12 mostra um exemplo das bases de dados *Two-spiral*, *Spiral* e *Half-rings*.

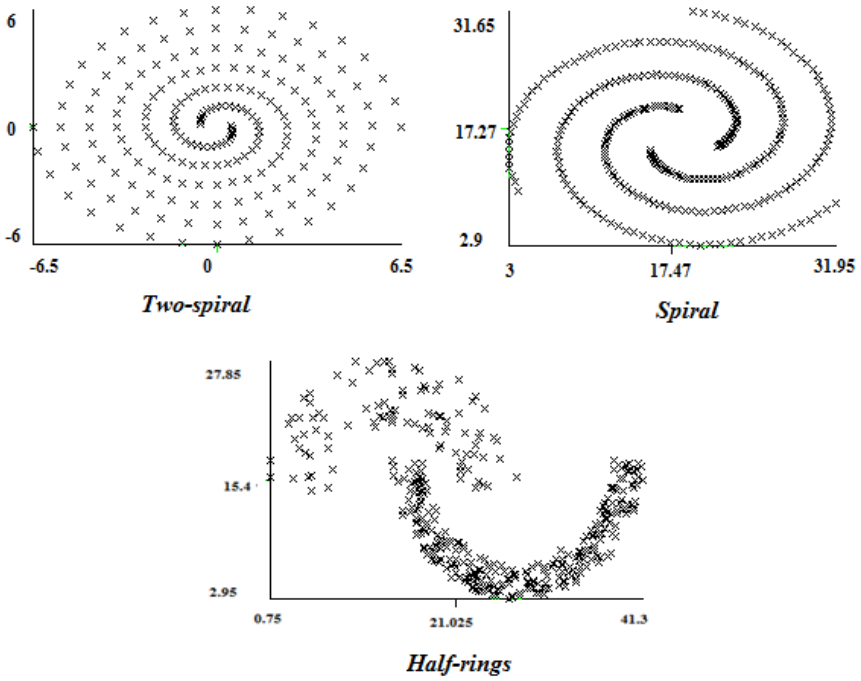

Figura 12 Exemplo das bases de dados, *Two-spiral*, *Spiral* e *Half-rings*, respectivamente. Imagen gerada pelo Weka

A Tabela 3 resume em detalhes as bases de dados utilizadas.

Tabela 3 Descrição da base de dados

Dados	***Nº de atributos***	***Nº de classes***	***Nº de instâncias por classe***	***Nº total de instâncias***
Iris	4	3	50	150
Diabetes	8	2	268-500	768
Yeast	8	10	5-463	1484
Two-Spiral	2	2	97-96	193
Spiral	2	3	101-106	312
Half-rings	2	2	97-276	373

4.1.2 Construção do *ensemble* e critério de avaliação

Para construir o *ensemble*, foram usados cinco algoritmos de clusterização, que foram: *K-means*, com a função de similaridade Euclidiana (KM-E) (MACQUEEN et al., 1967), *K-means*, com a função de similaridade *Manhattan* (KM-M) (VAIDYA; CLIFTON, 2003), o algoritmo de agrupamento *Expectation-maximization* (EM) (ZHANG; HSU; DAYAL, 2013), clusterização hierárquica (HC) (ACHTERT; BÖHM; KRÖGER, 2006) e PSO para clusterização, com a função de similaridade Euclidiana (COHEN; CASTRO, 2006; ESMIN; PEREIRA; ARAUJO, 2008; MERWE; ENGELBRECHT, 2003).

Antes de utilizar a função de consenso, duas novas bases de dados são formadas. Uma base de dados composta apenas dos rótulos dos dados resultantes da primeira fase do *ensemble* e outra composta dos dados originais acrescidas dos rótulos dos dados resultantes da primeira fase do *ensemble* de *clusters*, como descrito na seção 3.3. Em seguida, o processo de re-rotulagem é feito, usando os *clusters* criados, um algoritmo de *cluster* de referência escolhido a priori, como descrito na seção 3.2.

Com os dados re-rotulados, é possível aplicar a função de consenso. Para função de consenso, foram utilizados quatro algoritmos que são: *K-means*, com a função de similaridade Euclidiana, *K-means*, com a função de similaridade *Manhattan*, algoritmo de clusterização *Expectation-maximization* (EM), PSO para clusterização.

Para cada conjunto de dados, o PSO foi executado 150 vezes, usando 20 partículas, com os parâmetros $w = 0,17$, $w_{min} = 0,02$, c_1 e $c_2 = 0,2$, os mesmos utilizados em Esmin, Pereira e Araujo (2008) e Merwe e Engelbrecht (2003), segundo o qual os parâmetros garantem uma boa convergência.

A avaliação do desempenho dos algoritmos foi feita, comparando os resultados dos algoritmos de clusterização com os *clusters*, previamente, conhecidos de cada base de dados. A melhor correspondência possível dos *clusters* fornece uma medida do desempenho expressa em uma taxa de erro.

4.1.3 Resultados

O desempenho dos algoritmos de clusterização em cada base de dados é mostrado na Tabela 4, em que o melhor resultado encontrado entre os algoritmos (menor taxa de erro) está destacado com fonte em negrito. O algoritmo EM apresentou melhor resultado para metade das bases, porém seu erro médio foi superior ao algoritmo HC, que apresentou o menor erro médio graça ao resultado obtido na base de dados *Spiral*. O PSO, para clusterização, apresentou menor taxa de erro para duas bases e o terceiro menor erro médio.

Tabela 4 Resultado dos algoritmos de clusterização, taxa de erro (%)

Dados	***KM-E***	***KM-M***	***EM***	***HC***	***PSO***
Iris	11,33	10,66	9,33	34,00	**4,00**
Diabetes	33,20	34,76	33,98	34,76	**31,25**
Yeast	61,38	64,08	**58,15**	68,26	70,35
Two-Spiral	47,15	48,70	**47,15**	49,74	47,67
Spiral	65,70	66,02	64,74	**0,00**	53,21
Half-rings	11,79	17,96	**8,84**	25,73	17,16
Erro médio	38,425	40,36	37,03	**35,41**	37,27

A Tabela 5 apresenta o desempenho dos algoritmos de clusterização atuando como funções de consenso em uma base de dados composta apenas por rótulos. O PSO destacou-se por apresentar a menor taxa de erro para quatro bases e a menor taxa de erro médio. O algoritmo HC não foi usado como função de consenso por apresentar o valor da taxa de erro discrepante em relação aos outros algoritmos.

Tabela 5 Resultados dos algoritmos de consenso na base de dados formada apenas por rótulos, taxa de erro (%)

Dados	***KM-E***	***KM-M***	***EM***	***PSO***
Iris	**10,00**	10,66	10,66	10,67
Diabetes	**32,94**	**32,94**	33,07	**32,94**
Yeast	71,42	68,12	**66,84**	68,53
Two-Spiral	49,74	49,74	49,74	**49,22**
Spiral	51,92	65,38	51,92	**42,95**
Half-rings	11,79	11,79	18,63	**0,27**
Erro médio	37,96	39,77	38,47	**34,09**

A Tabela 6 apresenta o desempenho dos algoritmos de clusterização atuando como funções de consenso em uma base de dados composta pelos dados usados na clusterização acrescidos dos rótulos. O algoritmo PSO apresentou o melhor resultado, obtendo a menor taxa de erro e menor erro médio para todas as bases de dados.

Tabela 6 Resultados dos algoritmos de consenso na base de dados formada pelos dados acrescentados rótulos, taxa de erro (%)

Dados	***KM-E***	***KM-M***	***EM***	***PSO***
Iris	10,66	**10,00**	10,66	**10,00**
Diabetes	32,94	32,94	32,94	**31,64**
Yeast	68,39	67,38	66,91	**65,16**
Two-Spiral	49,74	49,74	49,74	**47,67**
Spiral	51,92	51,92	66,02	**44,55**
Half-rings	11,79	14,2	15,01	**10,72**
Erro médio	37,57	37,69	40,21	**34,95**

A Figura 13 resume os resultados de todos os algoritmos nas duas fases do *ensemble* de *clusters*, apresentando a taxa de erro médio de cada algoritmo. É possível comparar lado a lado o desempenho de cada algoritmo atuando como algoritmo de clusterização e nos dois formatos de base de dados usados na função de consenso.

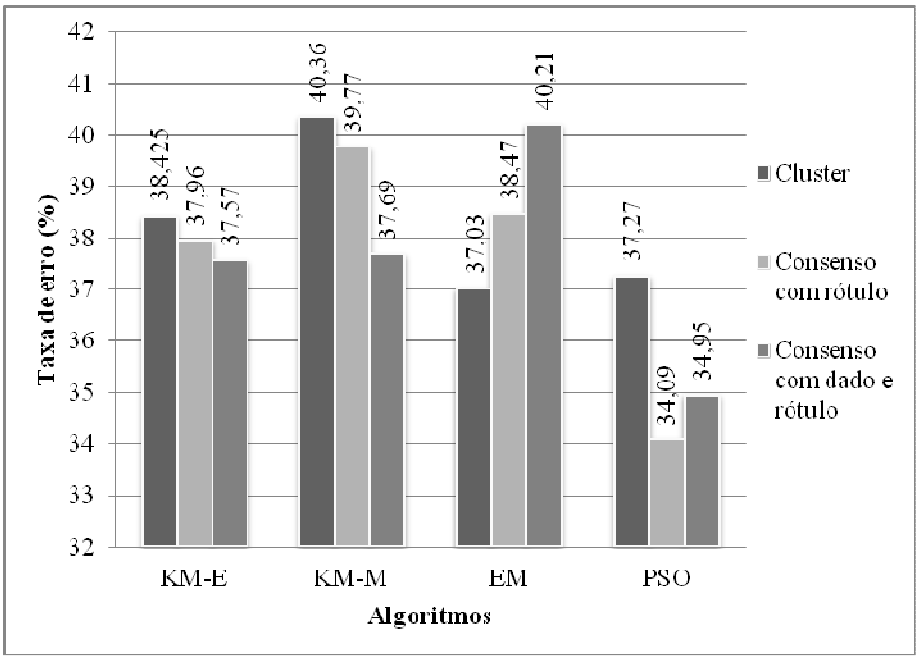

Figura 13 Erro médio de cada algoritmo sendo usado como algoritmo de clusterização e função de consenso do experimento 1

Dentre os algoritmos de consenso, o algoritmo PSO apresenta o melhor resultado em ambos os formatos das bases de dados. Os melhores resultados foram apresentados quando o PSO atuava somente sobre o rótulo dos dados, em virtude do espaço de busca limitado que o favorecia. O algoritmo *K-means*, com a função de similaridade Euclidiana, e o *K-means,* com a função de similaridade *Manhattan* apresentaram uma pequena melhoria em seus resultados, essa melhoria foi acentuada na base de dados composta pelos dados acrescido dos rótulos. O *Expectation-maximization* apresentou piora de seu resultado, quando utilizado como função de consenso, essa piora se tornou mais evidente quando adicionados os rótulos aos dados.

4.1.4 Considerações

Nesse experimento, o PSO foi aplicado como um método no *ensemble* e, também, como função consenso. O estudo empírico compara a precisão do algoritmo proposto com outros algoritmos, em seis bases de dados e em duas diferentes estruturas para os dados usados pela função de consenso: somente os rótulos e os dados acrescidos dos rótulos. Pelos resultados experimentais infere-se que o algoritmo PSO, usado como função de consenso, apresentou o melhor resultado, produzindo *clusters* tão bons ou melhores que os outros algoritmos usados na função de consenso.

4.2 Experimento 2 - PSO com diferentes medidas de similaridade aplicado ao ensemble de *clusters*

Para esse experimento, cinco versões do algoritmo PSO com diferentes medidas de similaridade foram empregadas nas duas fases do *ensemble* de *clusters*. O intuito desse experimento foi o de verificar o quanto as diferentes medidas de similaridade influenciam o comportamento do PSO no *ensemble*. Na primeira fase, cinco diferentes versões do PSO participam do *ensemble* junto com o *K-means* com a função de similaridade euclidiana. Todos os algoritmos usados na primeira fase, também, são usados na segunda fase como função de consenso. Para esse experimento, foram utilizadas seis bases de dados do repositório de referência da UCI e três dessas bases são reais e outras três sintéticas.

4.2.1 Base de dados

Os experimentos foram feitos usando bases de dados artificiais e reais. Foram usados seis conjuntos de dados, três deles reais *Iris*, *Diabetes* e *Wine*, obtidos do repositório de referência da UCI e três bases de dados sintéticas, *two-spiral*, *spiral* e *half-rings*. A Tabela 7 resume em detalhes as bases de dados utilizadas.

Tabela 7 Descrição da base de dados

Dados	***Nº de atributos***	***Nº de classes***	***Nº de instâncias por classe***	***Nº total de instâncias***
Iris	4	3	50	150
Diabetes	8	2	268-500	768
Wine	14	3	48-71	178
Two-Spiral	2	2	97-96	193
Spiral	2	3	101-106	312
Half-rings	2	2	97-276	373

4.2.2 Construção do *ensemble* e critério de avaliação

Para construção do *ensemble*, foram utilizadas seis diferentes algoritmos de clusterização, que são: o PSO para clusterização, com cinco diferentes medidas de similaridade: distância Euclidiana, distância *Manhattan*, Distância Cosseno, Distância *Chessboard* e Distância Canberra (WOLFRAM, 2013), bem como o *K-means* com a função de similaridade Euclidiana. Cada algoritmo de clusterização irá gerar *clusters* para o mesmo conjunto de dados.

Antes de utilizar a função de consenso, duas novas bases de dados são formadas. Uma base de dados composta apenas dos rótulos dos dados resultantes

do *ensemble*, e outra composta dos dados originais acrescidas dos rótulos dos dados resultantes da primeira fase do *ensemble* de *clusters*.

Com re-rotulação dos dados, é possível aplicar a função de consenso. Para a função de consenso foram utilizados seis diferentes algoritmos, que são: K-means com a função de similaridade Euclidiana, e o PSO com cinco diferentes medidas de similaridade.

Foi feita a avaliação dos algoritmos de clusterização de duas maneiras. Na primeira, foi realizada a comparação dos resultados dos algoritmos de clusterização com os *clusters* previamente conhecidos de cada base de dados. Na segunda maneira, foi feita a medida da homogeneidade *intracluster* e a separabilidade *intercluster*. Um bom método de clusterização irá produzir *clusters* de alta qualidade com alta similaridade *intracluster* e baixa similaridade *intercluster*.

$$intracluster = \frac{\sum_{i=i}^{k} \sum_{x \in C_i} \|x - \mu_i\|^2}{N}$$

em que N é o número de instâncias, K é o número de *clusters*, e μ_i é o centroide do *cluster* C_i. Para a distância inter*cluster*, é feito o cálculo entre os centroides dos *clusters*, definido pela seguinte equação:

$$intercluster = min\left(\|\mu_i - \mu_j\|^2\right)$$

em que i =1, 2, 3, ..., K-1 e j = i+1, 2, 3, ..., K. A qualidade dos *clusters* criados depende tanto do valor de *intracluster* quanto de *intercluster*.

Para cada conjunto de dados, o PSO foi executado 150 vezes, usando 20 partículas, com os parâmetros $w = 0{,}17$, $w_{min} = 0{,}02$, c_1 e $c_2 = 0{,}2$, os mesmos

utilizados em Esmin, Pereira e Araujo (2008) e Merwe e Engelbrecht (2003), segundo o qual os parâmetros garantem uma boa convergência.

4.2.3 Resultados

A taxa de erro de algoritmos de clusterização em cada base de dados é mostrada na Tabela 8. O melhor resultado (a menor taxa de erro) é destacado com fonte em negrito. Os algoritmos PSO fornecem melhores resultados, exceto na base de dados *Wine* em que o algoritmo *K-menas* conseguiu melhores resultados.

Tabela 8 Resultado dos algoritmos de clusterização, taxa de erro (%)

Dados	**KM Euc.**	**PSO Euc.**	**PSO Man.**	**PSO Cos.**	**PSO Ches.**	**PSO Can.**
Diabetes	33,20	31,64	33,85	35,42	**30,86**	33,07
Wine	**5,61**	26,97	26,97	25,84	25,84	25,28
Iris	11,33	4,67	4,67	**2,67**	5,33	5,33
Spiral	65,70	64,10	61,86	**55,77**	58,65	60,26
Two spiral	47,15	47,67	46,11	45,60	45,60	**44,56**
Half-rings	11,79	15,82	13,14	12,87	14,21	**7,51**

Na Tabela 9, podemos notar a taxa de erro de algoritmos de clusterização, agindo como funções de consenso em cada base de dados, formada apenas pelos rótulos. O melhor resultado (a menor taxa de erro) está destacado com fonte em negrito.

Tabela 9 Resultados dos algoritmos de consenso na base de dados formada apenas pelos rótulos, taxa de erro (%)

Dados	**KM Euc.**	**PSO Euc.**	**PSO Man.**	**PSO Cos.**	**PSO Ches.**	**PSO Can.**
Diabetes	32,81	33,46	32,16	32,81	**31,12**	32,94
Wine	26,40	23,60	24,72	**13,48**	17,98	23,60
Iris	4,66	2,67	4,67	**2,00**	4,67	2,67
Spiral	63,46	59,29	58,01	58,65	57,37	**57,05**
Two spiral	46,11	45,60	45,60	44,04	**42,49**	**42,49**
Half-rings	**12,06**	12,87	12,87	**12,06**	**12,06**	**12,06**

Os algoritmos PSO, com diferentes medidas de similaridade, fornecem os melhores resultados em todos as bases de dados. O algoritmo *k-means* conseguiu o melhor resultado na base de dados *Half-rings* empatado com o algoritmo PSO.

Na Tabela 10, podemos notar a taxa de erro de algoritmos de clusterização, agindo como funções de consenso em cada base de dados, formada dos dados acrescidos dos rótulos. O melhor resultado (a menor taxa de erro) está destacado com fonte em negrito.

Tabela 10 Resultados dos algoritmos de consenso na base de dados formada dos dados acrescidos dos rótulos, taxa de erro (%)

Dados	**KM Euc.**	**PSO Euc.**	**PSO Man.**	**PSO Cos.**	**PSO Ches.**	**PSO Can.**
Diabetes	32,81	**30,34**	35,42	39,97	31,51	36,59
Wine	**21,34**	26,97	26,97	26,40	26,40	26,40
Iris	5,33	**2,67**	**2,67**	**2,67**	**2,67**	**2,67**
Spiral	63,46	62,18	58,33	60,26	60,58	**57,05**
Two spiral	46,11	47,15	46,63	45,08	46,11	**44,56**
Half-rings	12,06	12,87	14,75	10,72	11,80	**10,19**

O algoritmo PSO, com medida de similaridade Canberra, forneceu quatro melhores resultados, e todos os algoritmos PSO tiveram a mesma taxa de erro na base de dados *Iris*.

Na Tabela 11, estão descritos os valores de *intracluster* dos algoritmos de clusterização, agindo como funções de consenso em cada base de dados, formada dos dados acrescidos dos rótulos. O melhor resultado (a menor valor) está destacado com fonte em negrito.

Tabela 11 Valores de *intracluster*, base de dados formada dos dados acrescidos dos rótulos

Dados	**KM Euc.**	**PSO Euc.**	**PSO Man.**	**PSO Cos.**	**PSO Ches.**	**PSO Can.**
Diabetes	6,265	4,854	5,650	5,163	**4,717**	5,450
Wine	17,38	17,45	**16,05**	16,18	16,06	16,22
Iris	**0,114**	0,130	0,132	0,123	0,135	0,124
Spiral	0,686	**0,659**	0,945	0,944	0,684	0,856
Two spiral	**25,30**	31,38	32,80	30,58	30,04	35,23
Half-rings	**0,588**	0,632	0,627	0,659	0,683	0,849

O algoritmo *k-means*, com medida de similaridade euclidiana, fornece três melhores resultados. Apesar dos algoritmos PSO não definirem mais melhores resultados, os valores alcançados estão próximos do melhor resultado.

Na Tabela 12, podemos notar os valores de *intercluster*, os melhores valores (valores mais altos) estão destacados com fonte em negrito.

Tabela 12 Valores de *intercluster*, base de dados formada dos dados acrescidos dos rótulos

Dados	**KM Euc.**	**PSO Euc.**	**PSO Man.**	**PSO Cos.**	**PSO Ches.**	**PSO Can.**
Diabetes	1,673	1,732	1,732	1,556	**2,449**	1,718
Wine	2,237	**2,693**	2,645	1,771	1,708	1,800
Iris	2,442	2,645	2,449	**1,672**	2,645	2,633
Spiral	**2,700**	1,182	1,231	2,425	1,732	2,006
Two spiral	**1,743**	1,170	1,732	1,357	1,407	1,356
Half-rings	2,409	2,639	**2,645**	2,488	2,449	**2,645**

Os algoritmos PSO forneceram quatro melhores resultados, cada algoritmo com um melhor resultado. O algoritmo *k-means* definiu seus dois melhores resultados.

A Figura 14 resume os resultados de todos os algoritmos nas duas fases do *ensemble* de *clusters*, apresentando a taxa de erro médio de cada algoritmo. É possível comparar lado a lado o desempenho de cada algoritmo atuando como algoritmo de clusterização e nos dois formatos de base de dados usados na função de consenso.

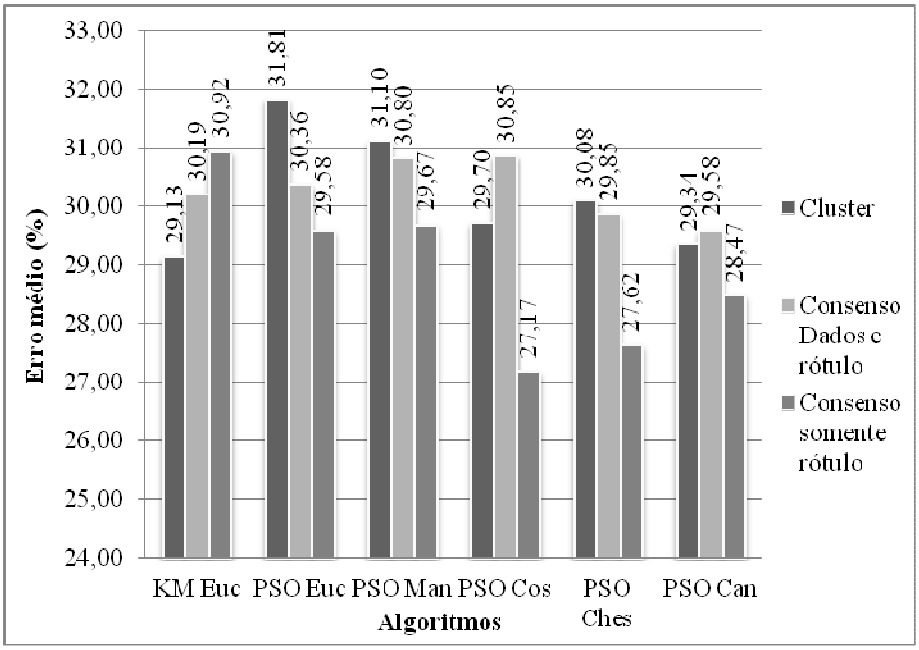

Figura 14 Erro médio de cada algoritmo sendo usado como algoritmo de clusterização e função de consenso do experimento 2

4.2.4 Considerações

Esse experimento apresentou a aplicação de PSO para resolver o problema do *ensemble* de *clusters*. O PSO foi aplicado como método de clusterização e, também, como função consenso. A taxa de erro dos algoritmos PSO, com cinco diferentes medidas de similaridade, foi comparado com o algoritmo de clusterização *K-means* em seis bases de dados como *benchmark*. Pelos resultados deduz-se que o algoritmo PSO *ensemble* possui baixa taxa de erro e produz bons *clusters*. Também foram avaliados os valores de *intracluster* e *intercluster* como critério de qualidade dos *clusters* criados, e verificou-se que os *clusters* criados pelo PSO com diferentes medidas se similaridades são tão bons quanto os criados pelo *K-means*.

4.3 Experimento 3 - Aplicação do *ensemble* com PSO para predição de defeitos em *software*

Nesse experimento, o *ensemble* de *clusters* foi aplicado ao estudo de predição de defeitos em software. O número de defeitos remanescentes em um software proporciona uma visão sobre a qualidade do mesmo. Sistemas de detecção de defeitos predizem defeitos usando métricas de software e técnicas de mineração de dados. A análise de *clusters* tem sido adotada para construção de modelos de software para predição de defeitos. Nesse experimento, o algoritmo PSO em *ensemble* é proposto para melhorar a qualidade da predição. Por um estudo empírico mostra-se que o PSO pode ser uma boa escolha para ser usado na construção de um modelo de software para predição de defeitos.

4.3.1 Descrição do problema de defeitos em *software*

A equipe de desenvolvimento de software tenta aumentar a qualidade do software diminuindo o tanto quanto possível o número de defeitos. É impossível eliminar completamente os defeitos de um software, mas se tenta minimizá-los ao máximo. Um defeito em software é um erro ou falha que produz um resultado incorreto ou inesperado, fazendo com que ele se comporte de forma inesperada. Os defeitos em software não são apenas um componente inerente do produto de software, são fatores significativos da qualidade do software (GRAY et al., 2011; INGLE; DESHPANDE, 2013).

Um sistema de predição de defeito em software é responsável por localizar módulos defeituosos em um software. Ele ajuda a melhorar a qualidade do software por meio da construção de modelos preditivos que permitem a identificação de módulos defeituosos. O sistema de predição de defeito pode classificar os módulos defeituosos em níveis de defeito ou por categoria, defeituoso ou não defeituoso (KARTHIK; MANIKANDAN, 2010).

Nesse experimento, foi feito um estudo empírico que compara o algoritmo PSO aplicado ao *ensemble* de *clusters* frente a outros algoritmos de clusterização na tarefa de predição de defeitos em software. Foi feita uma avaliação da precisão desse método para ser aplicado na construção de um sistema de predição de defeito em software.

4.3.2 Base de dados

Nessa seção, será descrita a base de dados usada nesse experimento. Uma descrição da *Metrics Data Program* (MDP) (BOETTICHER; MENZIES;

OSTRAND, 2007; MENZIES et al., 2012), diversas métricas usadas para predição de defeito de software.

A PROMISE (PRedictOr Models In Software Engineering) *Software Engineering Repository* disponibilizou publicamente o projeto NASA IV&V *Metrics Data Program* desenvolvida pela Galaxy Global Corporation, Inc. da NASA. Essa base de dados contém métricas de software e os dados de erro associados em nível de função/método dos projetos da NASA de desenvolvimento de software.

O repositório se encontra disponível para download no repositório da PROMISE *data*. A Tabela 13 exibe as métricas e uma descrição de cada uma das 22 métricas. Cada métrica equivale um atributo na base de dados, sendo a métrica *Problems* o atributo de classe.

Tabela 13 Métricas

	Métrica	Significado
1	Loc	McCabe's "linhas de código"
2	v(g)	McCabe "complexidade ciclomática"
3	ev(g)	McCabe "complexidade essencial"
4	iv(g)	McCabe "complexidade do projeto"
5	N	Halstead "operadores totais + operandos"
6	V	Halstead "volume"
7	L	Halstead "comprimento do programa"
8	D	Halstead "dificuldade"
9	I	Halstead "inteligência"
10	E	Halstead "esforço"
11	B	Halstead "erro"
12	T	Halstead's "tempo estimado"
13	loCode	Halstead's "quantidade de linhas"
14	loComment	Halstead's "quantidade de linhas de comentários"
15	loBlank	Halstead's "quantidade de linhas em branco"
16	loCodeAndComments	Quantidade de linhas de código e comentários
17	uniq_Op	operadores únicos
18	uniq_Opnd	operandos únicos
19	total_Op	operadores totais
20	total_Opnd	total de operandos
21	BranchCount	Quantidade de ramificações
22	Problems	O módulo apresenta ou não problemas relatados.

Na Tabela 14 são apresentadas as bases de dados utilizadas neste trabalho. O número de módulos é a quantidade de instâncias pertencentes àquela base de dados.

Tabela 14 Base de dados

Projeto	**Nº de Módulos**	**Defeito**	**Linguagem**
CM1	506	9.5%	C
KC1	2108	15.4 %	C++
KC2	522	20.5 %	C++

4.3.3 Construção do *ensemble* e critério de avaliação

Para construir o *ensemble*, foram usados cinco algoritmos de clusterização, que são: *K-means* com a função de similaridade Euclidiana (KM-E), *K-means* com a função de similaridade Manhattan (KM-M), o algoritmo de clusterização *Expectation-maximization* (EM) e o PSO para clusterização com as funções de similaridade Euclidiana (PSO-E) e Manhattan (PSO-M). Esses algoritmos de clusterização irão gerar seus próprios *clusters* utilizando o mesmo conjunto de dados.

Após a criação dos *clusters* pelo *ensemble*, é construído um novo conjunto de dados composto apenas dos rótulos, que será utilizado pela função de consenso. Em seguida, o processo de re-rotulagem é feito nesse novo conjunto de dados, de acordo com o algoritmo de clusterização de referência escolhido *a priori*.

Com os dados re-rotulados, é possível aplicar a função de consenso. Para a função de consenso, foram utilizados os mesmos cinco algoritmos anteriormente usados na clusterização.

A avaliação do desempenho dos algoritmos foi realizada comparando os resultados dos algoritmos de clusterização com os *clusters*, previamente, conhecidos de cada base de dados. A melhor correspondência possível dos *clusters* fornece uma medida do desempenho expressa em uma taxa de erro.

Para cada conjunto de dados, o PSO foi executado 150 vezes, usando 20 partículas, com os parâmetros $w = 0,17$, $w_{min} = 0,02$, $c_1 = c_2 = 0,2$. Esses parâmetros garantem uma boa convergência e foram utilizados por Esmin, Pereira e Araujo (2008) e Merwe e Engelbrecht (2003).

4.3.4 Resultados

A Tabela 15 mostra a taxa de erro (%) dos algoritmos de clusterização em cada base de dados. O melhor resultado (menor taxa de erro) para cada base está destacado com fonte em negrito. O algoritmo PSO-E conseguiu o melhor desempenho dentre os algoritmos no *ensemble*, seguido pelo KM-E, ao analisar o erro médio e as bases de dados com melhor resultado.

Tabela 15 Taxa de erro (%) obtido por cada algoritmo de clusterização

Base de Dados	KM-E	KM-M	EM	PSO-E	PSO-M
cm 1	16,06	18,67	29,71	**10,04**	28,71
kc 1	19,01	22,99	27,54	**15,41**	15,60
kc 2	19,92	27,39	30,84	20,69	**19,54**
Erro médio	18,33	32,01	29,36	**15,38**	21,28

A Tabela 16 apresenta o desempenho dos algoritmos de clusterização atuando como funções de consenso em cada base de dados. A taxa de erro que apresentou redução, em relação à Tabela 15, é acompanhada pelo símbolo ↓, e a que apresentou aumento pelo símbolo ↑. Como os algoritmos atuam na nova

base de dados constituída apenas dos rótulos dos dados, os valores encontrados tendem a ser mais homogêneos.

Tabela 16 Taxa de erro (%) obtida por cada algoritmo atuando como função de consenso

Base de Dados	KM-E	KM-M	EM	PSO-E	PSO-M
cm 1	29,71 ↑	18,67	29,71	18,67 ↑	**10,04** ↓
kc 1	27,54 ↑	**22,99**	**22,99** ↓	23,00 ↑	23,00 ↑
kc 2	**20,30** ↑	**20,30** ↓	30,84	20,31 ↓	20,31 ↑
Erro médio	25,85 ↑	20,65 ↓	27,84 ↓	20,66 ↑	**17,78** ↓

O algoritmo PSO-M, atuando como função de consenso, conseguiu o melhor desempenho ao analisar somente o erro médio, seguido pelo KM-M e PSO-E. O algoritmo EM apresentou o pior resultado nas duas fases do *ensemble* de *clusters*, apesar de ter seu erro médio reduzido.

Comparando as Tabelas 15 e 16, os algoritmos de clusterização quando usados como função de consenso tiveram um aumento na taxa de erros em 7 análises (resultados com ↑), manteve a taxa em 4 análises e melhorou sua taxa em 4 análises (resultados com ↓). Apesar dos algoritmos apresentarem aumento de erros na maioria das análises, três dos cinco algoritmos tiveram seu erro médio diminuído. A média do erro dos algoritmos na fase de *ensemble* que foi de 23,27% caiu para 22,55% quando os algoritmos atuaram como função de consenso.

Os algoritmos que apresentaram piora ao fim do *ensemble* de *clusters*, foram aqueles que utilizaram a medida de similaridade Euclidiana, já os que utilizaram a medida de similaridade Manhattan apresentaram melhora. A Figura 15 resume os resultados de todos os algoritmos nas duas fases do *ensemble* de

clusters. É possível comparar lado a lado o desempenho de cada algoritmo atuando como algoritmo de clusterização e como função de consenso.

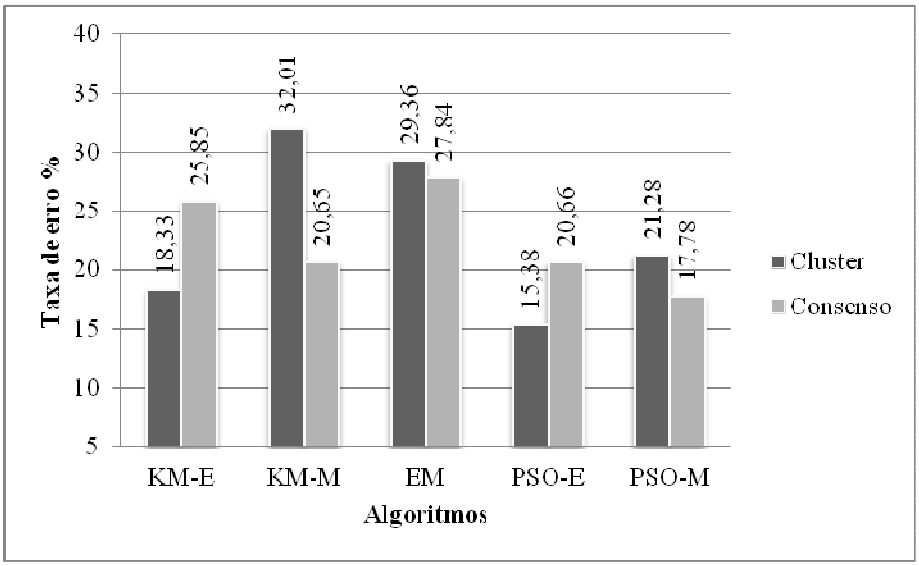

Figura 15 Erro médio do algoritmo de clusterização e função de consenso do experimento 3

Dentre os algoritmos escolhidos para esse experimento, o PSO apresentou os melhores resultados. Para construção de um sistema de predição de defeito em software, o PSO, utilizando a função de similaridade Manhattan, foi o que apresentou melhor desempenho.

4.3.5 Considerações

Esse experimento apresentou o desempenho do *ensemble* de *clusters* com o PSO como sugestão de algoritmo a ser aplicado na construção de um modelo de software para predição de defeitos em software. Nesse experimento, o PSO foi aplicado como um método no *ensemble* e, também, como função

consenso. O estudo empírico comparou a precisão do algoritmo proposto com outros algoritmos de consenso, em três bases de dados usadas para predição de defeitos da NASA. Os resultados experimentais mostraram que dentre os algoritmos, o PSO, com a medida de similaridade Manhattan, apresentou *clusters* tão bons ou melhores que os outros algoritmos usados na função de consenso.

5 CONCLUSÃO E CONSIDERAÇÕES FINAIS

Esse trabalho apresentou a aplicação do algoritmo *Particle Swarm Optimization* ao *ensemble* de *clusters*. O método do *ensemble* de *clusters* combina múltiplos *clusters* gerados por diferentes algoritmos de clusterização em uma solução de *cluster* único. Nesse trabalho, o processo consiste de duas partes: um *ensemble* e uma fase de re-rotulação e função de consenso.

O PSO foi aplicado como um método no *ensemble* e, também, como função consenso. Foram realizados três experimentos, no primeiro, o PSO foi empregado nas duas fases do *ensemble* de *clusters*, a fim de verificar o comportamento do PSO frente aos algoritmos *K-means* com a função de similaridade Euclidiana, *K-means* com a função de similaridade *Manhattan* e o algoritmo de clusterização *Expectation-maximization* (HC). O PSO mostrou os melhores resultados com as menores taxas de erro. No segundo experimento, o algoritmo PSO foi usado com cinco diferentes medidas de similaridade e o algoritmo *K-means* com a função de similaridade *Manhattan* nas duas fases do *ensemble* de *clusters*. Além da análise sobre a percentagem de erro cometido ao clusterizar a base de dados, também, foram calculados os valores de *intracluster* e *intercluster* a fim de investigar a qualidade dos *clusters* criado. O PSO apresentou as menores taxas de erro ao fim do *ensemble,* criando *clusters* de qualidade.

Já no terceiro experimento, o *ensemble* de *clusters* com o PSO foi aplicado ao estudo de predição de defeitos em *software*. O PSO junto com *K-means* com a função de similaridade Euclidiana, *K-means* com a função de similaridade Manhattan e o algoritmo de clusterização *Expectation-maximization* foram usados nas duas fases do *ensemble* de *clusters*. A base de dados usada para esse experimento contém métricas de software e os dados de erro associados em nível de função/método dos projetos da NASA de

desenvolvimento de software. Nesse experimento, o PSO, também, apresentou os melhores resultados dentre os algoritmos utilizados, é o mais indicado caso utilizado na construção de um sistema de predição de defeito em software.

Os resultados mostram que o algoritmo PSO, usado nas duas fases do *ensemble* de *clusters*, apresentou o melhor resultado, produzindo resultados tão bons ou melhores que os outros algoritmos. O PSO manteve os bons resultados independente da forma como os dados eram passados para a segunda fase do *ensemble*, sejam os rótulos ou os dados acrescidos dos rótulos.

Em trabalhos futuros, pretende-se investigar o uso do PSO aplicado ao *ensemble* de *clusters* para trabalhar com dados de alta dimensão. O desafio de clusterizar dados de alta dimensão surgiu nestes últimos anos e, em particular, os algoritmos de clusterização convencionais não apresentam bons resultados sobre esse tipo de base de dados. Variações do PSO adaptadas para lhe dar com dados de alta dimensionalidade devem ser investigadas e introduzido no contexto do ensemble de clusters.

REFERÊNCIAS

ACHTERT, E.; BÖHM, C.; KRÖGER, P. Deli-clu: boosting robustness, completeness, usability, and efficiency of hierarchical clustering by a closest pair ranking. In: _____. **Advances in knowledge discovery and data mining**. Berlin: Springer, 2006. p. 119-128.

AHMADI, A.; KARRAY, F.; KAMEL, M. S. Flocking based approach for data clustering. **Natural Computing**, Berlin, v. 9, n. 3, p. 767-791, 2010.

ALAM, S.; DOBBIE, G.; RIDDLE, P. An evolutionary particle swarm optimization algorithm for data clustering. In: SWARM INTELLIGENCE SYMPOSIUM, 2008, Saint Louis. **Proceedings…** Saint Louis: IEEE, 2008. p. 1-6.

ASUNCION, A.; NEWMAN, D. J. **UCI machine learning repository**. Irvine: University of California, 2007. Disponível em: <http://www.ics.uci.edu/~mlearn/MLRepository.html>. Acesso em: 10 dez. 2013.

BOETTICHER, G.; MENZIES, T.; OSTRAND, T. **Promise repository of empirical software engineering data**. Morgantown: West Virginia University, 2007. Disponível em: <http://promisedata.org/data>. Acesso em: 10 dez. 2013.

CHANG, H.; YEUNG, D. Y. Robust path-based spectral clustering. **Pattern Recognition**, New York, v. 41, n. 1, p. 191-203, 2008.

COELHO, R.; ESMIN, A. A. A. Diferentes abordagens usando PSO para clusterização de dados por consenso. In: CONGRESSO BRASILEIRO DE INTELIGÊNCIA COMPUTACIONAL, 11., 2013, Recife. **Anais...** Recife: CBIC, 2013. 1 CD-ROM.

COHEN, S. C.; CASTRO, L. N. de. Data clustering with particle swarms. In: IEEE CONGRESS ON EVOLUTIONARY COMPUTATION, 2006, Vancouver. **Proceedings…** Vancouver: IEEE, 2006. p. 1792-1798.
COLE, R. M. **Clustering with genetic algorithms**. Peth: University of Western, 1998. 110 p.

ESMIN, A. A. A.; COELHO, R. Consensus clustering based on particle swarm optimization algorithm. In: IEEE INTERNATIONAL CONFERENCE ON SYSTEM, MAN, AND CYBERNETICS, 2013, Manchester. **Proceedings…** Manchester: IEEE, 2013. p. 2280-2285.

ESMIN, A. A. A.; COELHO, R.; MATWIN, S. A review on particle swarm optimization algorithm and its variants to clustering high-dimensional data. **Artificial Intelligence Review**, Amsterdam, v. 42, n. 1, p. 1-23, 2013.

ESMIN, A. A. A.; LAMBERT-TORRES, G. Application of particle swarm optimization to optimal power systems. **International Journal of Innovative Computing, Information and Control**, Fukuoka, v. 8, n. 3A, p. 1705-1716, 2012.

ESMIN, A. A. A.; PEREIRA, D. L.; ARAUJO, F. D. Study of different approach to clustering data by using the particle swarm optimization algorithm. In: IEEE WORLD CONGRESS ON COMPUTATIONAL INTELLIGENCE, 2008, Hong Kong. **Proceedings…** Hong Kong: IEEE, 2008. p. 1817-1822.

FERN, X. Z.; BRODLEY, C. E. Solving cluster ensemble problems by bipartite graph partitioning. In: INTERNATIONAL CONFERENCE ON MACHINE LEARNING, 21., 2004, New York. **Proceedings…** New York: ACM, 2004. p. 36.

GHAEMI, R. et al. A survey: clustering ensembles techniques. **World Academy of Science, Engineering and Technology**, Las Cruces, v. 3, n. 2, p. 636-645, 2009.

GHOSH, A.; JAIN, L. C. **Evolutionary computation in data mining**. Berlin: Springer, 2005. 266 p.

GRAY, D. et al. The misuse of the nasa metrics data program data sets for automated software defect prediction. In: ANNUAL CONFERENCE ON EVALUATION & ASSESSMENT IN SOFTWARE ENGINEERING, 15., 2011, Durham. **Proceedings…** Durham: IET, 2011. p. 96-103.

HALL, M. et al. The weka data mining software: an update. **ACM SIGKDD Explorations Newsletter**, New York, v. 11, n. 1, p. 10-18, 2009.

HAN, J.; KAMBER, M. **Data mining:** concepts and techniques. Burlington: M. Kaufmann, 2005. 800 p.

INGLE, P.; DESHPANDE, M. Software quality analysis with clustering method. **International Journal of Applied Information Systems**, New York, v. 5, n. 2, p. 8-10, 2013.

JAIN, A. K.; LAW, M. H. Data clustering: a user's dilemma. In: _____. **Pattern recognition and machine intelligence**. Berlin: Springer, 2005. p. 1-10.

JIA, J.; CHUA, H. C. Solving two-spiral problem through input data representation. In: IEEE INTERNATIONAL CONFERENCE ON NEURAL NETWORKS, 1995, Perth. **Proceedings...** Perth: IEEE, 1995. v. 1, p. 132-135.

KARTHIK, R.; MANIKANDAN, N. Defect association and complexity prediction by mining association and clustering rules. In: IEEE INTERNATIONAL CONFERENCE ON COMPUTER ENGINEERING AND TECHNOLOGY, 2., 2010, Chengdu. **Proceedings…** Chengdu: ICCET, 2010. p. 569.

KENNEDY, J.; EBERHART, R. Particle swarm optimization. In: IEEE INTERNATIONAL CONFERENCE ON NEURAL NETWORKS, 1995, Perth. **Proceedings...** Perth: IEEE, 1995. v. 4, p. 1942-1948.

KENNEDY, J. F.; KENNEDY, J.; EBERHART, R. C. **Swarm intelligence**. Burlington: M. Kaufmann, 2001. 512 p.

KRIEGEL, H. P.; KRÖGER, P.; ZIMEK, A. Clustering high-dimensional data: a survey on subspace clustering, pattern-based clustering, and correlation clustering. **ACM Transactions on Knowledge Discovery from Data**, New York, v. 3, n. 1, p. 1:1-1:58, 2009.

KUNCHEVA, L.; HADJITODOROV, S.; TODOROVA, L. Experimental comparison of cluster ensemble methods. In: IEEE INTERNATIONAL CONFERENCE ON INFORMATION FUSION, 9., 2006, Florence. **Proceedings…** Florence: IEEE, 2006. p. 1-7.

LI, T.; DING, C. Weighted consensus clustering. **SIAM**, Philadelphia, v. 1, n. 2, p. 798-809, 2008.

LUXBURG, U. V. A tutorial on spectral clustering. **Statistics and Computing**, London, v. 17, n. 4, p. 395-416, 2007.

MACQUEEN, J. et al. Some methods for classification and analysis of multivariate observations. In: SYMPOSIUM ON MATHEMATICAL STATISTICS AND PROBABILITY, 5., 1967, Berkeley. **Proceedings…** Berkeley: SMS, 1967. v. 1, p. 281-297.

MENZIES, T. et al. **The promise repository of empirical software engineering data**. Disponível em: <http://www.promisedata.googlecode.com>. Acesso em: 10 dez. 2012.

MERWE, D. van der; ENGELBRECHT, A. P. Data clustering using particle swarm optimization. In: IEEE CONGRESS ON EVOLUTIONARY COMPUTATION, 3., 2003, Canberra. **Proceedings…** Canberra: IEEE, 2003. v. 1, p. 215-220.

NISHA, M.; MOHANAVALLI, S.; SWATHIKA, R. Improving the quality of clustering using cluster ensembles. In: IEEE CONFERENCE ON INFORMATION & COMMUNICATION TECHNOLOGIES, 2013, Jeju Island. **Proceedings…** Jeju Island: IEEE, 2013. p. 88-92.

NGUYEN, N.; CARUANA, R. Consensus clusterings. In: IEEE INTERNATIONAL CONFERENCE ON DATA MINING, 7., 2007, Omaha. **Proceedings…** Omaha: IEEE, 2007. p. 607-612.

NIU, Y.; SHEN, L. An adaptive multi-objective particle swarm optimization for color image fusion. In: _____. **Simulated evolution and learning**. Berlin: Springer, 2006. p. 473-480.

OMRAN, M. G.; SALMAN, A.; ENGELBRECHT, A. P. Dynamic clustering using particle swarm optimization with application in image segmentation. **Pattern Analysis and Applications**, Berlin, v. 8, n. 4, p. 332-344, 2006.

SHI, Y.; EBERHART, R. C. Parameter selection in particle swarm optimization. In: _____. **Evolutionary programming VII**. Berlin: Springer, 1998. p. 591-600.

SILVA, A.; NEVES, A.; COSTA, E. Chasing the swarm: a predator prey approach to function optimisation. In: INTERNATIONAL CONFERENCE ON SOFT COMPUTING, 8., 2002, Brno. **Proceedings…** Brno: ICSC, 2002. 1 CD-ROM.

STREHL, A.; GHOSH, J. Cluster ensembles: a knowledge reuse framework for combining multiple partitions. **The Journal of Machine Learning Research**, Edinburgh, v. 3, p. 583-617, 2003.

TAN, P. N.; STEINBACH, M.; KUMAR, V. **Introduction to data mining**. Boston: A. Wesley Longman, 2005. 769 p.

TOPCHY, A.; JAIN, A. K.; PUNCH, W. Clustering ensembles: models of consensus and weak partitions. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, New York, v. 27, n. 12, p. 1866-1881, Dec. 2005.

TOPCHY, A.; JAIN, A. K.; PUNCH, W. A mixture model of clustering ensembles. In: CITESEER SIAM INTERNATIONAL CONFERENCE ON DATA MINING, 1., 2004, Lake Buena Vista. **Proceedings...** Lake Buena Vista: SIAM, 2004. p. 379-390.

VAIDYA, J.; CLIFTON, C. Privacy-preserving k-means clustering over vertically partitioned data. In: ACM SIGKDD INTERNATIONAL CONFERENCE ON KNOWLEDGE DISCOVERY AND DATA MINING, 9., 2003, New York. **Proceedings…** New York: ACM, 2003. p. 206-215.

WITTEN, D. M.; TIBSHIRANI, R. A framework for feature selection in clustering. **Journal of the American Statistical Association**, New York, v. 105, n. 490, p. 713-726, 2010.

WOLFRAM, R. **Distance and similarity measures**. Disponível em: <http://reference.wolfram.com/mathematica/guide/DistanceAndSimilarityMeasures.html>. Acesso em: 10 nov. 2013.

YANG, L. Y.; ZHANG, J. Y.; WANG, W. J. Cluster ensemble based on particle swarm optimization. In: WRI GLOBAL CONGRESS ON INTELLIGENT SYSTEMS, 9., 2009, Xiamen. **Proceedings…** Xiamen: WRI, 2009. v. 3, p. 519-523.

YANG, Y.; KAMEL, M. Clustering ensemble using swarm intelligence. In: IEEE SWARM INTELLIGENCE SYMPOSIUM, 3., 2003, New York. **Proceedings…** New York: IEEE, 2003. p. 65-71.

ZHANG, B.; HSU, M.; DAYAL, U. **K-harmonic means-a data clustering algorithm**. Disponível em: <http://www.hpl.hp.com/techreports/1999/HPL-1999-124.pdf>. Acesso em: 10 nov. 2013.

ZHAO, B.; GUO, C.; CAO, Y. A multiagent-based particle swarm optimization approach for optimal reactive power dispatch. **IEEE Transactions on Power Systems**, New York, v. 20, n. 2, p. 1070-1078, 2005.