

Diferentes Abordagens Usando PSO para Clusterização de Dados por Consenso

Rodrigo A. Coelho
Departamento de Ciência da Computação
Universidade Federal de Lavras - UFLA
Lavras, MG, Brasil
rodrigoac@comp.ufla.br

Ahmed. A. A. Esmin
Departamento de Ciência da Computação
Universidade Federal de Lavras - UFLA
Lavras, MG, Brasil
ahmed@dcc.ufla.br

Resumo— Clusterização é uma importante tarefa na mineração de dados e tem sido utilizada por muitos pesquisadores em diferentes áreas. O ensemble de clusters combina múltiplos clusters criados por diferentes algoritmos de clusterização em uma solução única de clusters, considerado uma alternativa robusta e precisa frente algoritmos individuais de clusterização. 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. Neste trabalho, o algoritmo Particle Swarm Optimization (PSO) é proposto para resolver o problema de função de consenso. Um estudo empírico de nossa proposta frente a outros algoritmos de função de consenso mostra que o PSO produz resultados tão bons ou melhores, mesmo utilizando duas diferentes estruturas de bases de dados para a função de consenso.

Palavras chave—componentes; Particle Swarm Optimization; clusterização; ensemble; função de consenso

I. INTRODUÇÃO

A mineração de dados é uma das etapas do processo de descoberta de conhecimento em bases de dados (Knowledge Discovery in Databases - KDD) [1]. A clusterização é uma das técnicas de mineração de dados que consiste no processo em encontrar grupos nos quais os seus elementos são similares entre si e diferentes dos elementos de outros grupos de acordo com uma medida de similaridade [1] [2].

Nos últimos anos, diversos algoritmos de clusterização foram desenvolvidos. Esses algoritmos podem ser classificados em duas categorias de acordo com o tipo de saída, ou seja, *hard* ou *soft*. Um algoritmo *hard* aloca cada instância em apenas um cluster. Um algoritmo *soft* atribui graus de associação a cada instância para cada cluster. O Algoritmo do tipo *soft* pode ser convertido para *hard* atribuindo a instância ao cluster com maior grau de associação [3].

Para melhorar a estabilidade e robustez das saídas da clusterização, são utilizados ensembles de cluster [4]. Em um ensemble de cluster, vários algoritmos de clusterização fornecem soluções para a tarefa de clusterização. Este modelo é mais preciso do que algoritmos de clusters individuais, pois, melhora o resultado compensando erros cometidos por algum

algoritmo de clusterização pela intervenção da solução correta de outros.

O ensemble de cluster pode trazer benefícios que vão além daqueles que um simples algoritmo de clusterização pode conseguir. O ensemble de cluster gera melhores clusters, encontra configurações de clusters inalcançáveis por um algoritmo de clusterização simples, são menos susceptíveis a ruídos, variações amostrais e são capazes de integrar soluções a partir de múltiplas fontes de dados ou atributos distribuídos [5].

O Particle Swarm Optimization (PSO) é um algoritmo estocástico de base populacional proposto por Kennedy e Eberhart [6], 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 [7] [8], e processamento de imagem [9] [10]. Além disso, o PSO provou ser competitivo frente a algoritmos genéticos em várias tarefas, principalmente na área de otimização [11] [12] [13].

Nesse trabalho, é proposto o algoritmo Particle Swarm Optimization para atuar nas duas fases de um ensemble de clusters. Acreditamos que o algoritmo PSO é extremamente eficaz para este problema, independente de como os dados são estruturados e passados para a função de consenso. Um estudo empírico compara a precisão do método proposto com outros métodos de clusterização usados como função de consenso. Os experimentos demonstram que o algoritmo PSO produz resultados tão bons ou melhores do que outros algoritmos nas duas diferentes estruturas de dados utilizadas nesse trabalho.

O restante do artigo está organizado da seguinte forma. A Seção 2 apresenta os conceitos de ensemble e a função de consenso. A Seção 3 fornece uma visão geral do PSO. A Seção 4 apresenta o algoritmo PSO para clusterização. A Seção 5 descreve a metodologia proposta e os resultados experimentais. As conclusões são apresentadas na seção 6.

II. ENSEMBLE E FUNÇÃO DE CONSENSO

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. 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, sendo este modelo mais preciso do que algoritmos de clusters individuais [14] [15].

O ensemble de clusters é composto de duas fases. Na primeira fase, o conjunto de algoritmos que compõem 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 as combina através de uma função de consenso produzindo clusters como saída final [15] [16]. As duas fases do ensemble são descritas formalmente a seguir.

Ensemble: Dado um conjunto de dados com n instâncias $X = \{x_1, x_2, \dots, x_n\}$, e um conjunto formado por r algoritmos de clusterização $E = \{a_1, a_2, \dots, a_r\}$, cada solução de a_i é representada por um conjunto disjunto de clusters representado por $\lambda_i = \{c_1, c_2, \dots, c_k\}$, 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 1 ilustra o ensemble de clusters, a função de consenso Γ combina os resultados λ_i de um conjunto de algoritmos de clusterização E , produzindo um resultado λ final.

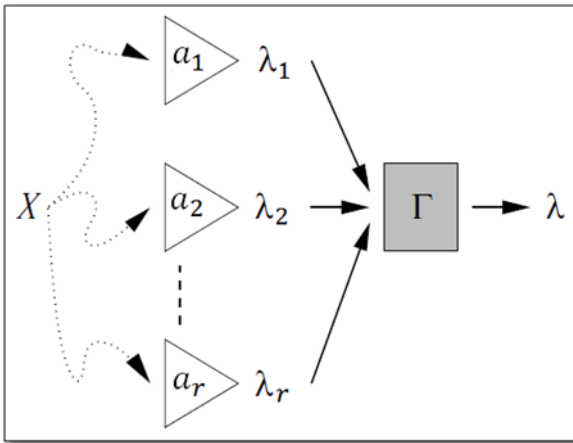


Fig. 1. Ensemble de clusters [14].

Rótulos: Dado um conjunto formado por r algoritmos de clusterização $E = \{a_1, a_2, \dots, a_r\}$, cada solução de a_i é representada por um conjunto disjunto de clusters representado por $\lambda_i = \{c_1, c_2, \dots, c_k\}$, cada conjunto λ_i poderá rotular os clusters criados de forma diferente, sendo necessário uma nova rotulação de acordo com uma λ_f de referência. A Tabela I exemplifica uma situação. A Tabela I exemplifica

uma base de dados com 6 instâncias, um ensemble com 4 algoritmos e seus resultados rotulados de forma diferente.

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-rotulação 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-rotulação de todo λ_i com o seu melhor acordo com uma λ_f de referência escolhida a priori [17].

TABELA I. 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

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

- *Hypergraph Consensus Methods:* em que os clusters podem ser representados como *hyperedges* em um grafo cujos vértices correspondem aos objetos a serem clusterizados.
- *Mutual Information Approach:* nessa abordagem, a função objetiva para um ensemble de clusters pode ser formulado como uma informação mútua entre a distribuição de probabilidade empírica dos rótulos na partição consenso e os rótulos no ensemble [5] [17].
- *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 [3] [4] [5].
- *Mixture Model Approach:* é um modelo probabilístico de consenso que utiliza uma mistura finita de distribuições multinomiais em um espaço de clusterização [15] [17].
- *Re-labeling Approach* ou *Voting Approach*, nesse método, todos os clusters do conjunto devem ser remarcados de acordo com um cluster de referência predeterminado [5] [17]. Depois do processo de re-rotulação, a abordagem por votação pode ser aplicada para determinar a associação entre cada instância de cluster.

III. O ALGORITMO PARTICLE SWARM OPTIMIZATION

O algoritmo Particle Swarm Optimization (PSO) é um método de otimização baseado na simulação do comportamento social de bando de pássaros. Geralmente enquadrado na computação evolutiva, o PSO tenta encontrar a solução ideal utilizando uma população de partículas [6] [18]. Alguns dos atrativos do PSO incluem a facilidade de implementação e o fato de que nenhuma informação de gradiente é necessária.

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 como se segue:

$$x_i(t+1) \leftarrow x_i(t) + v_i(t) \quad (1)$$

$$v_i(t+1) \leftarrow \omega v_i(t) + c_1 r_1 (pbest_i(t) - x_i(t)) + c_2 r_2 (gbest(t) - x_i(t)) \quad (2)$$

Nas equações (1) e (2), $x_i(t)$ 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 [7].

As equações (3) e (4) definem como os melhores valores pessoal e global são atualizados no tempo t , respectivamente. Suponha que o enxame consiste de s partículas.

Assim, $i \in 1 \dots s$

$$pbest_i(t+1) = \begin{cases} pbest_i(t) & \text{se } f(pbest_i(t)) \leq f(x_i(t+1)) \\ x_i(t+1) & \text{se } f(pbest_i(t)) > f(x_i(t+1)) \end{cases} \quad (3)$$

$$gbest(t+1) = \min\{f(y), f(gbest(t))\}, \quad (4)$$

em que $y \in \{pbest_0(t), pbest_1(t), \dots, pbest_s(t)\}$

O processo do PSO é mostrado a seguir.

```

Inicializar uma população de partículas  $i$ .
Enquanto(as condições de término não são verdadeiras){
  Para cada partícula  $i$  faça{
    Atualizar  $i$  de acordo com as equações (1) e (2).
    Calcular o  $fitness$ .
    Atualizar o  $pbest_i(t)$  e  $gbest_i(t)$ , de acordo com as equações (3) e (4).
  }
}

```

O algoritmo PSO é muito rápido, simples e fácil de compreender e aplicar. Também tem poucos parâmetros para ajustar [19]. 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 conjunto de dados grande e complexo.

IV. PSO PARA CLUSTERIZAÇÃO

A utilização do PSO para clusterização é apresentada em [20], 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 máximos de clusters que podem ser criados. A partícula x_i é construída da seguinte forma:

$$x_i = (m_{i,1}, \dots, m_{i,j}, \dots, m_{i,N_c})$$

Na qual N_c se refere ao número de clusters a serem criados em $m_{i,j}$ 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:

$$f = \frac{\sum_{j=1}^{N_c} [\sum_{\forall Z_p \in C_{i,j}} d(Z_p, m_{i,j}) / |C_{i,j}|]}{N_c} \quad (5)$$

Na qual i é a partícula, Z_p denota o p -ésimo elemento no vetor de dados e d é a distância Euclidiana 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 a seguir.

```

Inicializar uma população de partículas  $i$ .
Enquanto(as condições de término não são verdadeiras){
  Para cada partícula  $i$  faça{
    Para cada dado  $Z_p$ {
      Calcular  $d(Z_p, m_{i,j})$  para todos os centróides.
      Atribuir  $Z_p$  ao cluster  $C_{i,j}$ , tal que
       $d(Z_p, m_{i,j}) = \min_{c=1 \dots N_c} \{d(Z_p, m_{i,j})\}$ 
      Calcular o  $fitness$  da partícula de acordo com a equação (5).
    }
    Atualizar o  $pbest_i(t)$  e  $gbest_i(t)$ , de acordo com as equações (3) e (4).
    Atualizar  $i$  de acordo com as equações (1) e (2).
  }
}

```

V. EXPERIMENTOS, RESULTADOS E DISCUSSÃO

Nesta seção, primeiro estão descritos os algoritmos usados para criar o conjunto de cluster e, em seguida, as funções de consenso utilizadas nos experimentos.

Para construir o ensemble foram usados cinco algoritmos de clusterização, que são: *K-means* com a função de similaridade Euclidiano (KM-E) [21], *K-means* com a função de similaridade Manhattan (KM-M) [22], o algoritmo de agrupamento *Expectation-maximization* (EM) [23], clusterização hierárquica (HC) [24] e PSO para clusterização [20], [25], [26]. Esses algoritmos de clusterização irão gerar seus próprios clusters utilizando 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. Em seguida, o processo de re-rotulagem é feito usando um algoritmo de cluster de referência escolhido a priori.

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 Euclidiano, *K-means* com a função de similaridade Manhattan, algoritmo de clusterização *Expectation-maximization* (EM) e PSO para clusterização.

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.

Os experimentos foram feitos usando bases de dados artificiais e reais. Foram usados seis conjuntos de dados, três deles reais (*Iris*, *Diabetes* e *Yeast*) obtidos do repositório de referência da UCI [27] e três bases de dados sintéticas, *two-spiral* [28], *spiral* [29] e *half-rings* [30]. A Tabela II resume em detalhes as bases de dados utilizadas.

TABELA II. DESCRIÇÃO DAS BASES DE DADOS

Dados	Nº de atributos	Nº de classes	Nº de instâncias por classe	Nº total de instâncias
Iris	4	2	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

A Figura 2 mostra um exemplo da base de dados *half-rings*, base de dados de difícil clusterização, especialmente para algoritmos baseados em centroide.

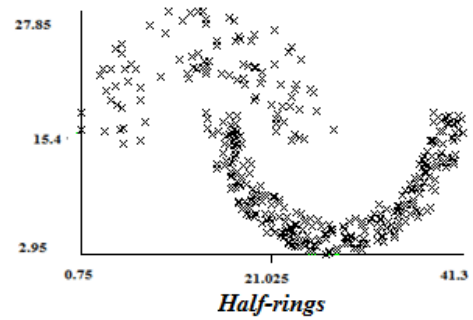


Fig. 2. Exemplo da base de dados *Half-rings*.

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 [20] [26], segundo o qual os parâmetros garantem uma boa convergência.

A Tabela III mostra o desempenho dos algoritmos de clusterização em cada base de dados, 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. O PSO para clusterização apresentou menor taxa de erro para duas bases e o terceiro menor erro médio.

TABELA III. 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 IV 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.

A Tabela V 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 IV. RESULTADO DOS ALGORITMOS DE CONSENSO NA BASE DE DADOS FORMADA SOMENTE 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

TABELA V. RESULTADO DOS ALGORITMOS DE CONSENSO NA BASE DE DADOS FORMADA PELOS DADOS ACRESCENTADO DE 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 3 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.

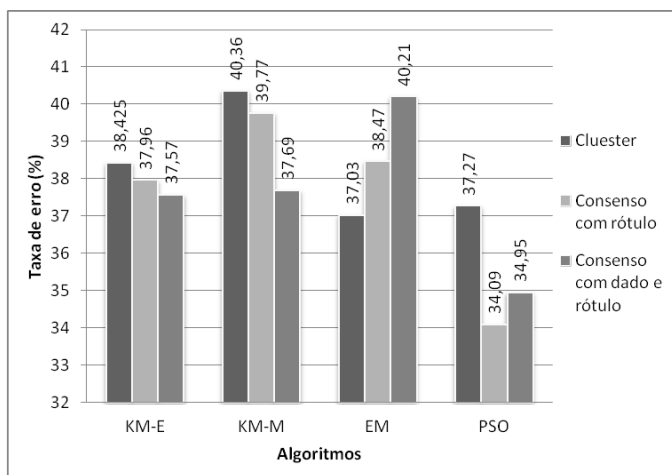


Fig. 3. Erro médio de cada algoritmo sendo usado como algoritmo de clusterização e função de consenso.

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, devido ao 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 adicionou os rótulos aos dados, aumentando a quantidade de dimensões da base de dados.

VI. CONCLUSÃO

Esse trabalho apresentou a aplicação do algoritmo Particle Swarm Optimization (PSO) para resolver o problema do ensemble de clusters e fez uma comparação utilizando dois tipos de bases de dados usadas na função de consenso. O método do ensemble de clusters combina múltiplos clusters gerados com diferentes algoritmos de clusterização em uma solução de cluster único. O processo consiste de duas partes: um ensemble e uma função de consenso. Nesse trabalho, o PSO é aplicado como um método no ensemble e também como função consenso. Um 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. Os resultados experimentais mostram 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.

VII. AGRADECIMENTOS

Os autores agradecem à Fapemig pelo apoio financeiro e à CAPES pela concessão da bolsa de estudo.

Referências

- [1] Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P., Uthurusamy, R. (1996), "Advances in Knowledge Discovery and Data Mining", Chapter 1, AAAI/MIT Press 1996
- [2] Jiawei, H., Micheline, K. (2001), "Data Mining, Concepts and Techniques", Morgan Kaufmann Publishers.
- [3] Li-ying Yang, Jun-ying Zhang, Wen-jun Wang. 2009. Cluster Ensemble Based on Particle Swarm Optimization. Intelligent Systems, 2009. GCIS '09. WRI Global Congress on. p.519-523
- [4] A. Topchy, A.K. Jain and W. Punch, Clustering Ensembles: Models of Consensus and Weak Partition," IEEE Trans. PAMI 2005, 27(12):1866-1881.
- [5] Nam Nguyen, Rich Caruana. 2007. Consensus Clusterings. Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on. p.607-612
- [6] Kennedy, J., Eberhart, R. C. (1995) "Particle Swarm Optimization", In Proceedings of IEEE Internal Conference on Neural Networks, Perth, Australia, Vol. 4, pp. 1942- 1948, 1995.

- [7] Alam, S., Dobbie, G., Riddle, P. (2008) "An Evolutionary Particle Swarm Optimization algorithm for data clustering", *Swarm Intelligence Symposium, 2008. SIS 2008. IEEE*, vol., no., pp.1-6, 21-23 Sept. 2008.
- [8] Esmine, A. A. A., Coelho R., Matwin, S. (2013) "A review on particle swarm optimization algorithm and its variants to clustering high-dimensional data", *Artificial Intelligence Review*, v. 1, p. 1-23, 2013. <http://dx.doi.org/10.1007/s10462-013-9400-4>.
- [9] Niu, Y., Shen, L. (2006) "An adaptive multi-objective particle swarm optimization for color image fusion", *Lecture notes in computer science, LNCS*. Pp 473-480.
- [10] Omran, M.G., Salman, A.A., Engelbrecht, A.P. (2006) "Dynamic clustering using particle swarm optimization with application in image segmentation", *Pattern Anal. Appl.* pp. 332-344.
- [11] Silva, A., Neves, A., Costa, E. (2002) "Chasing the Swarm: A Predator-Prey Approach to Function Optimisation", In *Proc. of the Mendel 2002 - 8th International Conference on Soft Computing*, pp. 103-110, Mendel 2002, Brno, Czech Republic.
- [12] Esmine, A.A.A.; Lambert-Torres, G. (2012) "Application of Particle Swarm Optimization to Optimal Power Systems", *International Journal of Innovative Computing, Information and Control (IJICIC)*, Vol. 8, No. 3 (A), pp 1705-1716.
- [13] Zhao, B., Guo, C. X., Cao, Y.J. (2005) "A Multiagent-based particle swarm optimization approach for optimal reactive power dispatch", *Power Systems. IEEE Trans. Power Systems*, vol. 20, no. 2, pp 1070-1078.
- [14] Strehl, A., Ghosh, J. (2003) "Cluster Ensembles - A Knowledge Reuse Framework for Combining Multiple Partitions", *J. Mach. Learn. Res.* 3 (March 2003), 583-617.
- [15] Ghaemi, R., Sulaiman, N., Ibrahim, H., Mustapha, N. (2009) "A Survey: Clustering Ensembles Techniques", *Proceedings of World Academy of Science, Engineering and Technology*, Vol. 38 (February 2009).
- [16] Fern, X. Z., Brodley, C. (2004) "Cluster ensembles for high dimensional clustering: An empirical study", *Journal of Machine Learning Research*.
- [17] Topchy, A., Jain, A. K., Punch, W. (2004) "A mixture model of clustering ensembles", In *Proceedings of the SIAM International Conference on Data Mining*.
- [18] Shi, Y., Eberhart, R.C. (1998) "Parameter selection in particle swarm optimization", *Evolutionary programming*, vol 1441 of *Lecture Note in computers science*. Springer, Berlin, pp 591-600.
- [19] Kennedy, J., Eberhart, R.C, Shi, Y. (2001) "Swarm Intelligence", Morgan Kaufmann, 2001.
- [20] Merwe, D.W., Engelbrecht, P. (2003) "Data Clustering using Particle Swarm Optimization", In *Congress on Evolutionary Computation, 2003. Proceedings of IEEE Congress on Evolutionary Computation 2003 (CEC 2003)*, Caribella: IEEE Computer Society, p. 215-220.
- [21] MacQueen, J. B. (1967) "Some Methods for classification and Analysis of Multivariate Observations", *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability*, Berkeley, University of California Press, 1:281-297.
- [22] Vaidya, J., Clifton, C. (2003) "Privacy-preserving k-means clustering over vertically partitioned data", In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 206-215). ACM.
- [23] Zhang, B., Hsu, M., Dayal, U. (1999) "K-harmonic means-a data clustering algorithm", *Hewlett-Packard Labs Technical Report HPL-1999-124*.
- [24] Achtert, E., Böhm, C., Kröger, P. (2006). "DeLi-Clu: Boosting Robustness, Completeness, Usability, and Efficiency of Hierarchical Clustering by a Closest Pair Ranking", *LNCS: Advances in Knowledge Discovery and Data Mining. Lecture Notes in Computer Science* 3918: 119-128.
- [25] Cohen, S. C. M., Castro, L. N. de. (2006) "Data Clustering with Particle Swarms. In: Congress on Evolutionary Computation", *Proceedings of IEEE Congress on Evolutionary Computation 2006 (CEC 2006)*. Vancouver: IEEE Computer Society, 2006. p. 1792- 1798
- [26] Esmine, A. A. A., Pereira., D. L., Araujo, F. P. A. (2008) Study of Different Approach to Clustering Data by Using The Particle Swarm Optimization Algorithm. In: *2008 IEEE Congress on Evolutionary Computation (IEEE CEC 2008)*, 2008, Hong Kong. *Proceedings of IEEE Congress on Evolutionary Computation, 2008*. p. 1817-1822.
- [27] Asuncion, A., Newman, D. J. (2007) "UCI Machine Learning Repository", <http://www.ics.uci.edu/~mllearn/MLRepository.html>. Irvine, CA: University of California, Department of Information and Computer Science
- [28] Jia, J., Chua, H-C. (1995) "Solving two-spiral problem through input data representation. *Neural Networks*", 1995. *Proceedings., IEEE International Conference on*. p. 132-135 vol.1.
- [29] Chang, H., Yeung, D.Y. (2008) "Robust path-based spectral clustering", *Pattern Recognition*, 2008. 41(1): p. 191-203.
- [30] Jain, A. K., Law, M. H. (2005). *Data clustering: A user's dilemma*. In *Pattern Recognition and Machine Intelligence, LNCS, V. 3776*, pp.1-10.