UMA ABORDAGEM EVOLUCIONÁRIA PARA A TAREFA DE AGRUPAMENTO DE DADOS

KELLY P. SILVA, RODRIGO G. F. SOARES, TERESA B. LUDERMIR, FRANCISCO A. T. CARVALHO

Centro de Informática, Universidade Federal de Pernambuco
Caixa Postal 7851, CEP 50732-970 Recife (PE), Brazil
{kps,rgfs,tbl,fatc}@cin.ufpe.br

Abstract— The grouping task consists of the discovery of interesting groups in a database. Such task is important and is widely studied in literature. In this work, we consider an evolutionary method for the attainment of cluster cohesion and spatially separated groups. The proposed algorithm (AgrEvo) uses a complete representation of the grouping, each partition is represented by a length-variable *chromosome*. The variation operators had been chosen in order to facilitate the exchange of information between groups. Two complementary objective functions were employed for the composition of the fitness function with the intention to find groups with arbitrary forms. A local search procedure that uses the algorithm *k*-means was employed to refine the obtained solutions. The proposed method does not require the setting of the number of clusters in advance. To evaluate the performance of the model, we use both real and simulated data.

Keywords— Evolutionary Algorithms, Clustering.

Resumo— A tarefa de agrupamento consiste na descoberta de grupos interessantes em uma base de dados. Tal tarefa é bastante importante e amplamente estudada na literatura. Neste trabalho, propomos um método evolucionário para a obtenção de grupos coesos e bem separados. O algoritmo proposto (AgrEvo) utiliza uma representação completa da solução, cada partição é representada por um *cromossomo* de tamanho variável. Os operadores de variação foram escolhidos de forma a facilitar a troca de informação entre grupos. Foram utilizadas duas funções objetivo complementares entre si para a composição da função de aptidão, com o intuito de encontrar grupos com formas arbitrárias. Um procedimento de busca local que utiliza o algoritmo *k*-médias foi empregado para refinar as soluções encontradas. No método proposto não há necessidade de ajuste do parâmetro do número de grupos *a priori*. Para avaliar o desempenho do modelo, utilizamos tanto dados reais quanto dados simulados.

Palavras-chave— Algoritmos Evolucionários, Agrupamento de Dados.

1 Introdução

O problema de agrupamento está inserido no contexto não-supervisionado de aprendizado de máquina, onde o aprendizado é direcionado aos dados, não requerendo conhecimento prévio sobre suas classes ou categorias [11]. Tal problema consiste em encontrar grupos em uma base de dados que tenham significado para o domínio dos dados estudados. Há na literatura uma grande variedade de algoritmos de agrupamento, cada um com suas características e peculiaridades. Cada algoritmo faz uso de uma heurística para encontrar o melhor agrupamento para um conjunto de dados. Uma das principais questões deste problema é encontrar grupos coesos e bem separados. A primeira meta é encontrar grupos cujas instâncias estejam mais próximas o possível entre si (próximas em termos de alguma medida de distância ou correlação). A segunda meta consiste na descoberta de grupos bem espaçados uns dos outros. Alguns algoritmos presentes na literatura tentam atingir uma ou outra meta, como é o caso do *k*-médias [13], que tenta formar grupos coesos (com mínima variância). Além disso, um algoritmo pode apresentar resultados melhores que outro, de acordo com a distribuição dos dados no espaço de atributos. Por exemplo, um algoritmo pode ser mais apropriado para encontrar grupos hiper-esféricos enquanto outro pode encontrar grupos com formas arbitrárias. Outra questão acerca do problema é que cada algoritmo pode encontrar estruturas com vários níveis de refinamento, dependendo dos valores de seus parâmetros. Por exemplo, o algoritmo *k*-médias encontra uma estrutura diferente para cada número de grupos (*k*). Porém, durante a análise exploratória dos dados, o número de grupos, geralmente, não é conhecido previamente, o que torna a tarefa mais complicada.

Existem várias técnicas de validação de agrupamentos na literatura, que podem auxiliar tanto na escolha do algoritmo mais apropriado quanto na seleção da melhor partição, dado que um algoritmo foi executado várias vezes com diferentes valores para seus parâmetros. Porém, a maioria das medidas de validação é tendenciosa [4], cada uma favorecendo um critério de agrupamento diferente. Daí surge outra questão concernente a agrupamento de dados: qual medida de validação deve ser utilizada para um dado conjunto de dados, uma vez que não se conhece o critério de agrupamento mais apropriado aos dados.

O espaço de busca do problema de agrupamento, dependendo do tamanho do conjunto de dados, pode ser muito grande. Daí, o tempo de execução de um algoritmo que busque todas as soluções nesse espaço pode se tornar proibitivo rapidamente. Os Algoritmos Evolucionários (AE's) são empregados para resolver problemas que são complexos, mal definidos e mal estruturados [2]. Tais características nos levam a empregar AE's em problemas de agrupamento.

A abordagem proposta é um Algoritmo Memético (AM) que evolui uma população de agrupamentos codificados em cromossomos de acordo com operadores especiais de variação. Esses operadores usam uma medida de interesse que guia o processo de re-

combinação e mutação para melhorar grupos que ainda não estão devidamente adequados. Além dos operadores, adotamos uma função de aptidão capaz de incorporar tanto a medida de coesão intra-grupos como a medida de separação inter-grupos. Para refinar a busca, lançamos mão de uma busca local baseada no algoritmo *k*-médias. E devido ao problema da rápida convergência dos AM's, aplicamos o compartilhamento de aptidão para manter a diversidade da população.

Aplicamos o algoritmo proposto a bases de dados reais e simulados a fim de verificarmos o comportamento do método em situações distintas.

2 Trabalhos Relacionados

Existem vários algoritmos de agrupamento na literatura. No entanto, vamos expor apenas dois deles que usam AE's.

O primeiro é o EvoCluster [8], que foca sua atenção em agrupamento de micro-arrays gênicos. Essa abordagem usa representação completa dos agrupamentos nos cromossomos, sua medida de aptidão consegue conduzir o processo de busca a características escondidas na base de dados gênicos, mesmo na presença de ruídos e dados faltosos. Tal método ainda introduz operadores de variação que serão usados no presente trabalho. Eles são operadores que levam em consideração a qualidade de cada gene, que codifica um determinado grupo. Eles facilitam a troca de informações relevantes entre grupos de dois agrupamentos e de grupos em um mesmo agrupamento. Os resultados para dados genéticos se mostraram bastante promissores, o que nos levou a empregar tais operadores em agrupamentos de outros contextos.

A segunda abordagem usa um Algoritmo Genético (AG) para evoluir uma população de agrupamentos oriundos de técnicas já existentes [3]. Esse método usa a técnica multi-objetivo NSGA II, ela tem como funções de aptidão a conectividade e a variância de um agrupamento. Essas medidas serão usadas neste trabalho para compor a função de aptidão. Tal abordagem também emprega um operador de recombinação especial, que usa a idéia de comitês de agrupamentos, para gerar a prole. O algoritmo citado teve um desempenho bastante superior ao dos algoritmos tradicionais.

3 Algoritmo Evolucionário

O AgrEvo codifica cada agrupamento em um cromossomo. Cada gene de um cromossomo representa um grupo e é composto por rótulos. Os rótulos identificam os registros do conjunto de dados.

A Figura 1 apresenta o esquema de codificação de um cromossomo. Este esquema apresenta um cromossomo que codifica *k* grupos, ou seja, ele possui *k* genes e o gene *i* apresenta n_i rótulos.

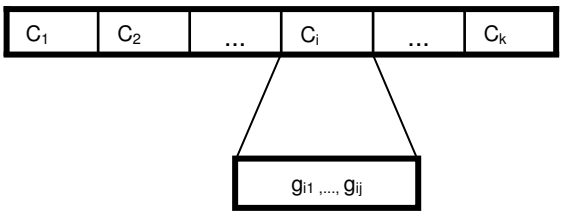

Figura 1 - Esquema de Codificação do Cromossomo.

A população inicial é gerada da seguinte forma: para cada indivíduo é escolhido aleatoriamente o número de grupos *k*, de um domínio que varia entre 2 e $\sqrt{n}$, onde *n* é o número de instâncias na base de dados. Em seguida, cada rótulo é associado a um desses grupos aleatoriamente.

O AgrEvo utiliza o método da roleta para a seleção dos pais. A cada geração, dois ou mais pais são selecionados para reprodução, sendo a probabilidade de seleção de cada indivíduo da população proporcional à sua aptidão.

O AgrEvo utiliza os operadores de cruzamento apresentados por [8]. Estes operadores foram especialmente projetados para facilitar a troca de informações entre dois cromossomos.

Foram utilizados dois operadores de cruzamento: um operador de cruzamento não guiado (UGC), onde a troca de informações entre os grupos ocorre de maneira aleatória e um operador de cruzamento guiado (GC), onde a troca de informações é guiada pelo grau de interesse de cada grupo. A medida de interesse associada a cada grupo será explicada na seção III-E.

Assumindo que os cromossomos P_1 e P_2 foram selecionados para recombinação e que o cromossomo P_1 codifica k_1 genes e o P_2, k_2 genes, isto é, o número de grupos codificados pelos indivíduos pode ser diferente. Assumindo também que MIN e MAX expressam, respectivamente, o número mínimo e máximo de genes que um cromossomo pode codificar, os seguintes passos são executados pelos operadores de cruzamento:

1. Selecionar a probabilidade P_g de um gene ser selecionado para cruzamento aleatoriamente de um domínio $[L_g,U_g]$, onde $0 \le L_g \le U_g \le 1$.
2. Selecionar a probabilidade P_r de um rótulo ser trocado por um outro de seu pai. Esse valor é escolhido aleatoriamente do domínio $[L_r,U_r]$, com $0 \le L_r \le U_r \le 1$.
3. Procedimento de Seleção de Genes
   a. Para o cruzamento não guiado (UGC), percorra todos os genes de P_1 e verifique, de acordo com a probabilidade P_g, se este será selecionado. O número de genes selecionados deve ser menor que o número de genes codificados pelos indivíduos k_1 e k_2;
   b. Para o cruzamento guiado (GC), baseado em P_g, determine N_g como sendo o número de genes que devem ser selecionados, onde $N_g < k_1,k_2$. Em seguida,

selecione os N_g genes mais interessantes de P_1.

4. Procedimento de substituição de rótulos
   a. Para o cruzamento não guiado (UGC), para cada gene selecionado de P_1, selecione aleatoriamente um gene de P_2 e utilizando um gerador de números randômicos percorra cada gene selecionado de P_1 para verificar quais rótulos serão substituídos, o número de rótulos selecionados é denotado por N_r. Os rótulos selecionados são removidos do gene. Em seguida, selecione aleatoriamente N_r, rótulos do gene de P_2 em questão e substitua os rótulos removidos.
   b. Para o cruzamento guiado (GC), comece com o gene mais interessante de P_1 e escolha o gene equivalentemente interessante de P_2, baseado na probabilidade P_r, percorra cada rótulo do gene de P_1 para selecionar aqueles que devem ser substituídos. Em seguida, substitua os rótulos selecionados por um mesmo número de rótulos selecionados aleatoriamente do gene em questão de P_2. Repita os passos acima para todos os genes selecionados de P_1.
5. Procedimento de Reparação
   Percorra todos os genes do primeiro filho para eliminar os rótulos duplicados. Devem ser removidos os rótulos duplicados dos genes que não sofreram substituição. Para os rótulos que não foram associados a nenhum gene, devem ser executados os seguintes procedimentos:
   a. Para o UGC, estes rótulos devem ser associados aleatoriamente a um dos genes;
   b. Para o GC, estes rótulos são reclassificados em um dos genes, utilizando o algoritmo de reclassificação apresentado na Seção III-D.
6. Repita os procedimentos de 1-5 com P_2, para produzir o segundo filho.

Depois do processo de cruzamento, os filhos gerados passam pelo operador de mutação, a fim de se conseguir diversidade e melhor explotação, além de ajudar o algoritmo a escapar de mínimos locais. Propomos o uso de seis diferentes tipos de operadores de mutação inspirados naqueles utilizados em [8]. Eles podem ser classificados como guiados e não guiados. Os operadores do primeiro tipo são aplicados aos genes seguindo a medida de interesse, enquanto os do segundo tipo escolhem os genes ao acaso. Em cada um desses tipos há uma sub-classificação: existem os operadores de remoção e reclassificação (guiado e não guiado), os operadores de mistura de grupos (guiado e não guiado) e os operadores de separação de grupos (guiado e não guiado). Os operadores de remoção e reclassificação removem rótulos de genes seguindo certo critério (guiado ou não guiado, mostrados a seguir) e em seguida os reclassificam utilizando um algoritmo de aprendizagem supervisionada (Naive Bayesian Learning, [13]) que será apresentado posteriormente. Os operadores de mistura de grupos (genes) escolhem genes seguindo certo critério e os une em um grupo. Já os operadores de divisão de grupos selecionam os genes que serão separados em dois grupos em um dado ponto. Daí, nós temos seis operadores de mutação.

Os primeiros operadores são de separação e mistura de genes, eles foram projetados especificamente para alterar o tamanho do cromossomo dinamicamente ao longo do processo evolutivo. A vantagem de tal característica é que não há a necessidade de se especificar o número de grupos a priori. O algoritmos dos seis operadores são descritos a seguir:

Operadores de mutação de remoção e reclassificação guiado (GRRM) e não guiado (UGRRM).

1. Selecione a probabilidade P_g de um gene ser selecionado
2. Selecione a probabilidade P_r de um rótulo em um gene ser removido
3. Procedimento de seleção de genes.
   a. Para o UGRRM, baseando-se na P_g, percorra cada gene no *cromossomo* para decidir se ele deve ser selecionado. O conjunto de genes selecionados é denotado por *Cs*.
   b. Para o GRRM, baseando-se em P_r, determine o número N_g de genes menos interessantes que devem ser selecionados. Então selecione os N_g genes menos interessantes, tal conjunto é denotado por *Cs*.
4. Procedimento de reparo de rótulos: para cada gene em *Cs*, baseando-se em P_r, percorra cada rótulo dos genes de *Cs* e selecione aqueles que devem ser removidos. Esses rótulos são representados por *Gs*.
5. Procedimento de reparo da prole: para aqueles rótulos que não estão associados a nenhum gene após suas remoções.
   a. Para o UGRRM, eles são associados aleatoriamente a algum dos genes.
   b. Para o GRRM, eles são reclassificados para algum dos genes usando o algoritmo de reclassificação descrito na próxima seção.

Operadores de mutação de mistura de genes guiado (UGMGM) e não guiado (GMGM).

1. Selecione a probabilidade P_g de um gene ser misturado.
2. Procedimento de seleção de genes.
   a. Para o UGMGM, baseando-se em P_g, percorra cada gene no cromossomo para selecionar aleatoriamente um gene para ser misturado. O conjunto dos genes selecionados é denotado por *Cs*.
   b. Para o GMGM, baseando-se em P_g, determine o número N_g de genes menos interessantes a serem misturados. Então selecione aqueles N_g menos interessantes. Tal conjunto de genes é denotado por *Cs*.

3. Procedimento de mistura de genes: para cada gene em *Cs*, selecione aleatoriamente um dos outros genes para ser misturado ao gene em questão. O número de genes após a mistura deve ser maior ou igual a *MIN*, caso contrário o operador termina.

Operadores de Mutação de Separação Guiado (UGSGM) e não guiado (GSGM).

1. Selecione a probabilidade P_g de um gene ser escolhido para ser dividido.
2. Procedimento de seleção de genes:
   a. Para o UGSGM, baseando-se em P_g, percorra cada gene no cromossomo para selecionar aleatoriamente um gene para ser dividido. O conjunto dos genes selecionados é denotado por *Cs*.
   b. Para o GSGM, baseando-se em P_g, determine o número N_g de genes não interessantes a serem divididos. Então selecione aqueles N_g menos interessantes. Tal conjunto de genes é denotado por *Cs*.
3. Procedimento de separação: para cada gene em *Cs*, divida-o aleatoriamente em dois genes. O número de genes resultante deve ser menor do que *MAX*, caso contrário o operador termina.

Para os operadores que usam o esquema de remoção e reclassificação, isto é, o GC e o GRRM, adotamos um esquema supervisionado para realocar os rótulos removidos. Os rótulos removidos são usados como instâncias de teste em um algoritmo de Aprendizagem Bayesiana [13] treinado com as instâncias cujos rótulos estão em um cromossomo, cada instância sendo associada a um grupo. Para o treinamento de tal algoritmo fazemos a correspondência entre grupo e classe, ou seja, o treinamento do algoritmo será feito seguindo a divisão das instâncias em grupos (classes) codificados nos genes. Assim, a instância é submetida a uma classificação, a classe obtida como resposta do algoritmo corresponderá ao grupo no qual a instância em questão será inserida. Essa abordagem busca guiar o processo de busca no sentido de encontrar padrões escondidos nos dados através da tentativa de colocar certas instâncias em grupos mais apropriados – esses novos grupos advêm de um algoritmo de aprendizagem supervisionada.

Uma função de aptidão que consiga avaliar as diferentes estruturas subjacentes nos dados é fundamental para que o Algoritmo Evolucionário proposto tenha êxito. Existem diversas medidas que avaliam um agrupamento, no entanto elas são tendenciosas [4] (atribuem maior qualidade a certos tipos de agrupamentos, por exemplo, agrupamentos com grupos hiper-esféricos). Daí a necessidade de se compor uma medida de aptidão robusta, que não apresente viés para algum tipo de forma de grupos. A seguir, descreveremos a função de aptidão proposta.

A variância intra-grupo [4] π de uma partição é calculada como a soma total das distâncias entre as instâncias e o centro dos seus grupos. Essa medida corresponde ao critério otimizado pelo algoritmo *k*-médias. Ela mede a qualidade de um agrupamento em termos da compactação (homogeneidade) de seus grupos. A Equação 1 denota a função de variância, onde μ_k é o centróide do grupo c_k e $d(.,.)$ é a função de distância euclidiana.

$$var(\pi) = \sum_{c_k \in \pi} \sum_{x_i \in c_k} d(x_i, \mu_k) \qquad (1)$$

Quanto menor o valor dessa medida, melhor a partição. Contudo, essa medida favorece grupos esféricos e não apresenta bons resultados para agrupamentos cujos grupos não estejam bem separados. O valor dessa medida melhora com o aumento do número de grupos.

A medida de conectividade reflete o grau com que instâncias vizinhas são colocadas em um mesmo grupo. A Equação 2 denota tal medida.

$$con(\pi) = \sum_{i=1}^{n} \sum_{j=1}^{v} a(x_i, nn_{ij}) \qquad (2)$$

$$a(x_i, nn_{ij}) = \begin{cases} \frac{1}{j}, \; se \; \neg \exists \, c_k : x_i, nn_{ij} \in c_k \\ 0, \; caso \; contrário \end{cases} \qquad (3)$$

Onde nn_{ij} é *j*-ésimo vizinho mais próximo da instância x_i, *n* é número de instâncias e *v* é a quantidade de vizinhos a serem considerados. Essa medida não apresenta restrições quanto à forma dos clusters, isto é, ela é apropriada para a avaliação de grupos de formas arbitrárias. O valor dessa medida melhora com a diminuição do número de grupos.

A função de aptidão proposta leva em consideração as medidas de variância e conectividade, que se complementam. A Equação 4 denota a função de aptidão.

$$f(\pi_{\mathbf{x}}) = \frac{1}{var(\pi_{\mathbf{x}}) \times con(\pi_{\mathbf{x}})} \qquad (4)$$

Onde $\pi_{\mathbf{x}}$ representa a partição codificada em um indivíduo $\boldsymbol{x}$. Essa medida incorpora duas funções objetivo que representam diferentes características em um agrupamento. Tal função não é restrita a nenhum domínio específico.

Os Algoritmos Genéticos simples mostraram ser bastante adequados para exploração do espaço de busca [2], no entanto, não conseguem realizar uma boa explotação na vizinhança das soluções. Daí, para melhorar o desempenho do método proposto, empregamos o algoritmo de agrupamento *k*-médias [13] como operador de vizinhança. A vizinhança de um dado indivíduo contém apenas um outro indivíduo, apenas executando o operador de movimento uma única vez.

A convergência prematura é um dos principais problemas dos AM's [2]. Então, para que haja a manutenção da diversidade na população, ou seja, soluções em vários nichos diferentes, um esquema de compartilhamento de aptidão [2] foi proposto. Com este método, os indivíduos têm sua aptidão proporcional à aptidão do nicho juntamente com a quantidade de indivíduos que se encontram naquela vizinhança. O compartilhamento da aptidão altera a aptidão do indivíduo $\boldsymbol{x}$ de acordo com a Equação 5, onde

sh(.) é uma função da distância d, n é o número de instâncias e σ_{share} é o raio em que os vizinhos do indivíduo x se encontram.

$$f'(x) = \frac{f(x)}{\sum_{j=1}^{n} sh(d(x, j))} \quad (5)$$

$$sh(d) = \begin{cases} 1 - (d / \sigma_{share}), \text{ se } d \le \sigma_{share} \\ 0, \text{ caso contrário} \end{cases} \quad (6)$$

A fim de assegurar que soluções boas não sejam descartadas, usamos seleção de sobreviventes *steady-state*, com os 4% piores indivíduos sendo substituídos pela prole.

4 Experimentos e Resultados

Para avaliar a performance do algoritmo evolucionário, utilizamos duas bases de dados artificiais apresentadas em [3]. A primeira base, que chamaremos Artificial 1, consiste de 588 registros, caracterizados por 2 atributos com valores no intervalo [0.0,1.0]. Essa base possui três possíveis estruturas com diferentes níveis de refinamento, podendo ter seus dados agrupados em 2, 5 ou 13 grupos. A segunda base, chamada Artificial 2, é formada por 485 registros, que também são caracterizados por 2 atributos e podem ser agrupados em dois diferentes níveis de refinamento, com 2 ou com 8 grupos, cujas estruturas podem ser visualizadas na Figura 2.

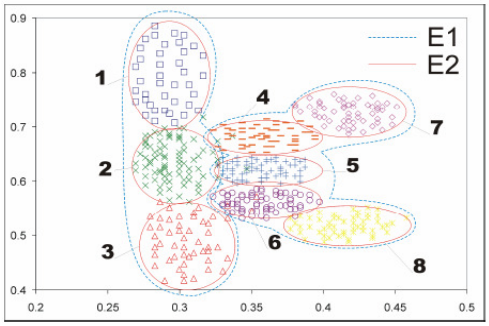

Figura 2 - Base Artificial 2 com 2 possíveis estruturas. Esta figura foi obtida em [3].

Além disso, o AgEvo também foi testado com outras bases de dados reais apresentadas em [12] – as bases Íris e Glass.

A validação das partições geradas pelo Agrevo foi baseada no índice Davies–Bouldin (DBI) [1] e no índice de Rand Corrigido (CR) [6].

O DBI é uma função que leva em consideração as distâncias inter e intra-grupos. O agrupamento é considerado de boa qualidade quando tem uma distância intergrupos relativamente grande e uma distância intra-grupos relativamente pequena. Utilizamos a distância Euclidiana como medida de distância. No caso do DBI, é considerado o melhor agrupamento aquele que possui o menor valor de DBI.

O CR é um índice de validação externa utilizado para comparar duas partições. Esse índice tem como vantagem não ser sensível ao número de grupos como outros índices tradicionais de validação externa [7]. Ele mede a similaridade entre dois agrupamentos, apresentando valor 0 quando as partições são aleatórias e valor 1 quando as partições casam perfeitamente. Em nosso caso, as partições obtidas pelo AgrEvo serão comparadas com as bases originais e o melhor agrupamento será aquele que apresentar o maior CR.

A efetividade do AgrEvo foi comparada com duas técnicas de agrupamento existentes na literatura: *k*-médias [9] e EM [10]. Para tanto, realizamos 30 execuções de cada técnica e apresentamos o melhor resultado para a medida CR e a média da medida DBI, obtidos por cada técnica para cada estrutura existente nas bases.

O AgrEvo foi executado 30 vezes para cada base. O tamanho da população foi de 50 indivíduos e o critério de parada foi o número de 50 gerações.

Para mostrar a performance do AgrEvo, escolhemos as soluções mais parecidas com cada uma das estruturas e calculamos os valores de CR. As Tabelas de 1 a 4 apresentam um resumo da comparação entre as diversas técnicas, apresentando os valores de CR e *k* obtidos para as bases Íris, Glass, Artificial 1 e 2, respectivamente.

A Tabela 5 apresenta os valores de média e desvio-padrão da medida DBI obtidos pelo AgrEvo, *k*-médias e EM para as diversas bases de dados.

Tabela 1. CR e *k* - base Íris.

Técnica	E1 (k=2)		E2 (k=3)	
	CR	*k*	CR	*k*
AgrEvo	1	2	0,66	3
k-médias	1	2	0,72	3
EM	0,35	5	0,51	5

Tabela 2. CR e *k* - base Glass.

Técnica	E1 (k=2)		E2(k=5)		E3(k=6)	
	CR	*k*	CR	*k*	CR	*k*
AgrEvo	0,98	2	0,7	4	0,51	6
k-médias	0,59	2	0,42	2	0,2	2
EM	0,45	5	0,43	5	0,24	5

Tabela 3. CR e *k* – base Artificial 1.

Técnica	E1 (k=2)		E2 (k=5)		E3 (k=13)	
	CR	*k*	CR	*k*	CR	*k*
AgrEvo	1	2	0,88	5	0,58	13
k-médias	1	2	0,84	8	0,54	9
EM	0,31	7	0,83	7	0,64	15

Tabela 4. CR e *k* - base Artificial 2.

Técnica	E1 (k=2)		E2 (k=8)	
	CR	*k*	CR	*k*
AgrEvo	0,96	2	0,7	9
k-médias	0,44	2	0,7	8
EM	0,23	7	0,68	7

Tabela 5 - Valores de DBI obtidos pelos diversos métodos.

Técnica	Íris		Glass		Artificial 1		Artificial 2	
	M	DP	M	DP	M	DP	M	DP
AgrEvo	0,52	0,23	1,27	0,35	0,78	0,087	0,78	0,08
k-médias	0,56	0,21	1,14	0,26	0,8	0,12	1,43	0,61
EM	0,96	0,07	1,42	0,18	0,85	0,07	0,87	0,09

A Figura 3 exibe a visualização do resultado obtido pelo AgrEvo para a base Artificial 2. Esta solução apresenta nove grupos e possui CR igual a 0,7, em relação à estrutura com 8 grupos previamente conhecida.

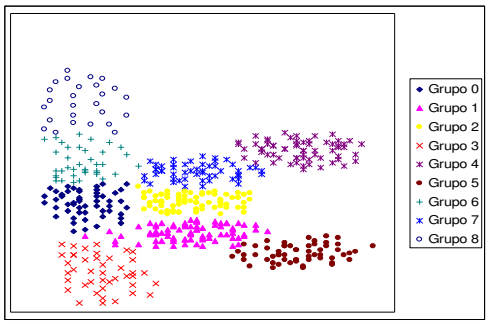

Figura 3 - Resultado obtido pelo AgrEvo para a base Artificial 2.

Podemos notar que para a base Íris, o AgrEvo conseguiu achar o melhor resultado em termos de CR para estrutura com 2 grupos, apenas o *k*-médias conseguiu melhor CR para a estrutura E2. Para o DBI, o AgrEvo obteve resultado superior aos das outras técnicas, sendo estatisticamente superior ao EM (através teste de hipótese com grau de confiança de 95%). Para a base Glass, o AgrEvo foi superior às outras técnicas para todas as estruturas consideradas (em termos do CR). O método proposto foi superado apenas pelo *k*-médias, na medida DBI.

Para a base Artificial 1, a abordagem proposta conseguiu encontrar a estrutura E1, teve o melhor CR para a E2 e foi superada pelo *k*-médias em E3. Além disso, foi superior em termos da média do DBI. Para a última base, considerando o CR, o AgrEvo foi superior para ambas as estruturas, além de ser estatisticamente superior às outras duas técnicas (com grau de confiança de 95%), para a medida DBI.

5 Conclusões

A tarefa de agrupamento de dados é bastante complexa, pois envolve uma série de questões como o formato dos grupos presentes nos dados, o número de grupos, a medida de validação adequada, entre outros. Neste trabalho, propomos um algoritmo evolucionário, denominado AgrEvo, com a finalidade de resolver os problemas citados acima.

O AgrEvo codifica uma partição em um único cromossomo, onde cada gene codifica um grupo. Baseando-se nessa estrutura, o algoritmo faz uso de operadores de variação para trocar e inserir informações nos cromossomos. Além disso, o algoritmo utiliza um procedimento de busca local, para melhorar o processo de explotação, e faz uso de um método de compartilhamento de aptidão, para garantir a manutenção da diversidade das soluções.

O AgrEvo foi testado em bases de dados artificiais e reais, mostrando-se adequado aos dois contextos. O método conseguiu agrupar os dados em estruturas com diferentes níveis de refinamento e apresentou melhores resultados que algoritmos de agrupamento como o *k*-médias e o EM, principalmente com o aumento da complexidade da estrutura presente na base de dados.

Os experimentos mostraram que o AgrEvo é bastante eficiente para a tarefa de agrupamento de dados, podendo ser aplicado a quaisquer conjuntos de dados sem restrições referentes ao domínio de conhecimento ou ao formato da estrutura que pode ser encontrada nos dados. Além disso, o método não necessita do conhecimento prévio do número de grupos presentes nos dados e apresentou resultados estáveis nas execuções realizadas.

Referências Bibliográficas

[1] Davies, D. L. and D. W. Bouldin (1979). A cluster separation measure. IEEE Trans. Pattern Anal. Machine Intell., vol. 1, no. 2, pp. 224–227.

[2] Eiben, A. E. and J.E. Smith (2003). Introduction to Evolutionary Computing, Springer.

[3] Faceli, Katti (2006). Um framework para análise de agrupamento baseado na combinação multi-objetivo de algoritmos de agrupamento. Universidade de São Paulo, USP, Brasil.

[4] Handl, J., J. Knowles, and D. Kell (2005). Computational cluster validation in post-genomic data analysis. Bioinformatics 21 (15), pp. 3201–3212.

[5] Handl, J. and J. Knowles (2007). An Evolutionary Approach to Multiobjetive Clustering. IEEE Transactions on Evolutionary Computation. Vol 11. pp. 56-76.

[6] Hubert, L. J. and P. Arabie (1985). Comparing partitions. Journal of Classification 2, pp. 193–218.

[7] Jain, A. and R. Dubes (1988). Algorithms for Clustering Data. Prentice Hall.

[8] Ma, P.C.H., Chan, K.C.C. Xin Yao Chiu, D.K.Y (2006). An Evolutionary Clustering Algorithm for Gene Expression Microarray Data Analysis. IEEE Transactions on Evolutionary Computation.

[9] MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. In Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics, pp. 281–297.

[10] McLachlan, G. and T. Krishman (1997). The EM algorithm and extensions. Wiley Series in Probability and Statistics. New York, NY: John Wiley & Sons.

[11] Mitchell, T. (1997). *Machine Learning*, McGraw-Hill.

[12] Newman, D., S. Hettich, C. Blake, and C. Merz (1998). UCI repository of machine learning databases. http://www.ics.uci.edu/~mlearn/ [Acessado em 30/03/2007]. University of California, Irvine, Dept. of Information and Computer Sciences.

[13] Witten, Ian H. and Eibe Frank (2005). Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations.