Otimização por Colônia de Abelhas Aplicada ao Problema de Seleção de Atributos

Daiany F. Lara, Andres J. Porfirio, Aurora T. R. Pozo

Departamento de Informática
Universidade Federal do Paraná (UFPR) – Curitiba, PR - Brasil
{daypicada, auroratrinidad}@gmail.com, andresjesse@yahoo.com.br

Abstract. *Work with large amounts of data is often difficult and computationally expensive, the feature selection is used to reduce databases in order to eliminate irrelevant information and preserve the most important. This paper presents the use of the bees algorithm (bee colony optimization - BCO) to the problem of features selection. The results presented in this paper as low computational cost, a smaller number of selected attributes (that resulted in better rates of correct answers in the classifier without loss of information) clearly demonstrates the effectiveness of the method.*

Resumo. *Trabalhar com grandes quantidades de dados muitas vezes é difícil e computacionalmente caro, a seleção de atributos é utilizada para reduzir bases de dados de forma a eliminar informações irrelevantes e preservar as mais importantes. Este trabalho apresenta a utilização do algoritmo das abelhas (bee colony optmization - BCO) para o problema de seleção de atributos. Os resultados apresentados neste trabalho demonstram claramente a eficácia do método, uma vez que pode-se verificar o baixo custo computacional e redução no número de atributos selecionados, os quais resultaram em melhores taxas de acertos no classificador sem perda de informações.*

1. . Introdução

O avanço da tecnologia fez com que o armazenamento de grandes quantidades de dados aumentasse, desta forma dificultando sua análise [Castanheira 2008]. Para amenizar este problema, surgiu no processo de Descoberta de Conhecimento ou KDD (*Knowlege Data Discovery*) a mineração de dados, que tem como objetivo destacar informações úteis em uma grande coleção de dados.

O processo KDD, divide-se em várias etapas: seleção, pré-processamento, extração de padrões e pós-processamento. Na etapa de pré-processamento acontece a seleção de atributos, que é essencial para o bom desempenho da mineração de dados. Esta seleção tem como intuito preparar, reduzir e transformar os dados a serem classificados, podendo ser aplicada na identificação dos atributos mais importantes de acordo com alguma métrica.

O objetivo deste trabalho é melhorar o desempenho do classificador na fase de mineração de dados. Para isso, foi utilizada uma técnica de seleção de atributos baseada na meta-heurística das abelhas BCO (*Bee Colony Optimization*). O BCO tem como objetivo selecionar os atributos mais relevantes da base.

O artigo está organizado da seguinte forma: seção 2 discute os principais conceitos sobre seleção de atributos. As medidas de avaliação utilizadas são descritas na seção 3, a seção 4 apresenta a aplicação do algoritmo BCO no problema de seleção de atributos. Na seção 5 são mostrados os resultados obtidos na implementação, e por fim, na seção 6, são relatadas as considerações finais.

2. . Seleção de Atributos

A tarefa de seleção de atributos é uma estratégia para lidar com um grande número de atributos armazenados em uma base de dados quando existe a necessidade de aplicação de um algoritmo de mineração de dados, mais especificamente uma técnica de classificação.

Independente do número de dados ou atributos, a seleção de atributos é umas das principais tarefas do pré-processamento, na qual é feita a preparação dos dados a serem minerados. De maneira geral nem todos os atributos são necessários para discriminar a classe de maneira precisa, e a utilização destes atributos podem gerar resultados imprecisos na etapa de classificação [Pereira 2009].

Além do objetivo de identificar os atributos relevantes e com informações essenciais, as técnicas de seleção de atributos são importante também para o melhor desempenho do classificador, redução e simplificação do conjunto de dados e principalmente, maior agilidade na tarefa de classificação [Pereira 2009].

As estratégias de avaliação do subconjunto de atributos, nada mais são do que as maneiras de medir a importância dos mesmos para obtenção do resultado final. Estas estratégias estão divididas em três diferentes abordagens: Embedded [Soares 2010], Wrapper [Spolaôr 2010] e Filter.

Este trabalho utiliza apenas a abordagem Filter, onde seleciona-se o subconjunto de atributos na fase de pré-processamento, independentemente do classificador. Esta abordagem pode avaliar os atributos individualmente e escolher os mais relevantes, além de avaliar diversos subconjuntos de atributos de forma heurística em busca de uma combinação que proporcione melhores resultados de classificação [Pila 2005].

3. . Medidas de Avaliação

A seleção dos atributos importantes em uma base de dados visa maximizar a separação dos exemplos de classes diferentes e minimizar a separação dos exemplos que pertencem a uma mesma classe [Lee 2005]. A seguir serão apresentadas as métricas pertencentes a medida de distância utilizada neste trabalho:

- **Distância Inter-Classe (DEC)**: Mensura a distância média entre cada centro de classe (a classe corresponde a um exemplo em que cada atributo e obtido pela média dos valores da classe em questão) e o centro do conjunto total dos dados [Spolaôr 2010]. Esta medida está representada na Equação 1:

$$DEC = \frac{1}{e}\sum_{cl=1}^{b} e_{cl} d\left(\vec{c}_{cl}, \vec{c}\right)$$

Equação 1: Fórmula para o cálculo da DEC.

Onde: ***e*** corresponde ao total de instâncias da base, ***b*** é o número de classe, $\vec{c}$ representa a média geral da base (soma dos valores de todos os atributos dividido pelo número de instâncias da base), $\overrightarrow{c_{cl}}$ média da classe (centro da classe) e, por fim, ***d*** faz o cálculo da distância euclidiana entre $\overrightarrow{c_{cl}}$ e c, onde, a média de cada classe até ***b*** e diminuída pela média geral da base, elevado ao quadrado.

- **Chi-Square**: Avalia a qualidade de um atributo de acordo com a sua correlação com a classe por meio de um teste estatístico X^2. Para cada valor a_i do atributo A ($1 \leq i \leq k$) e para cada valor c_j da classe C ($1 \leq j \leq m$), existe uma frequência esperada quando (A = a_i) e (C = c_j) que pode ser calculada pela fórmula representada na Equação 2:

$$e_{ij} = \frac{count\left(A = a_i\right) * count\left(C = c_j\right)}{N}$$

Equação 2: Cálculo da frequência utilizado em Chi-Square.

Onde: ***N*** é o número de instâncias, ***count(A = a_i)*** é o número de vezes que ocorre o valor de a_i do atributo A, e, ***count (C = c_j)*** é o número de instâncias que pertencem à classe c_j. A partir da frequência esperada de todas as combinações pode-se calcular a métrica Chi-Square pela fórmula apresentada na Equação 3:

$$X^2 = \sum_{i=1}^{k}\sum_{j=1}^{m}\left[\frac{o_{ij} - e_{ij}}{e_{ij}}\right]$$

Equação 3: Fórmula para o cálculo da Chi-Square.

Onde: o_{ij} é a frequência observada da combinação (A = a_i) e (C = c_j) [Pereira 2009].

- **. Otimização por Colônia de Abelhas**

Otimização por colônia de abelhas é uma meta-heurística pertencente ao grupo de técnicas de inteligência, bio inspirados, e também baseado na concepção construtiva [Teodorovic, Markovic e Orco 2006]. O método BCO baseia-se na maneira como as abelhas reais atuam em busca do alimento. A principal ideia do algoritmo BCO é criar uma colônia de abelhas artificial capaz de resolver eficientemente problemas de otimização combinatória através da simulação do comportamento das abelhas reais.

No início do processo de busca todas as abelhas artificiais estão localizadas na colmeia e se comunicam diretamente durante esse processo. Cada abelha realiza uma série de movimentos locais construindo uma solução para o problema. Passo a passo, as abelhas adicionam componentes à solução parcial até que se obtenha uma solução viável, ou seja, um caminho completo até a fonte de alimento [Teodorovic e Orco 2005].

Ao final da busca por fontes de alimentos, as abelhas campeiras (aquelas que saem em busca de novas fontes de alimento) retornam à colmeia e trocam informações com as abelhas seguidoras (responsáveis pela extração do alimento), a fim de informá-las sobre a qualidade, quantidade e localização da fonte encontrada. Essa troca de informações é realizada através de uma dança, chamada de *waggle dance*. As abelhas seguidoras, ao receberem tais informações, podem optar por seguir a abelha campeira que está realizando a dança [Lucic et Al 2006].

Sempre que uma abelha campeira retorna à colmeia, podem ocorrer três situações: (a) A abelha descarta a localização da comida e torna-se uma seguidora. (b) A abelha mantém o comportamento de campeira, descobrindo novas fontes, mas sem recrutar o resto da colonia ou ainda (c) A abelha pode recrutar outras abelhas antes de retornar à localização da fonte. As abelhas então optam por uma das alternativas citadas com base em uma determinada probabilidade. O recrutamento entre as abelhas se dá sempre em função da qualidade da fonte de alimento [Pham et Al 2006].

Existem duas fases distintas que constituem uma única etapa do algoritmo BCO, elas são: passo a frente (*Forward pass*) e passo a trás (*Backward pass*). Em cada passo a frente, todas as abelhas visitam n soluções componentes, gerando soluções parciais, e depois retornam a colmeia. O número de soluções componentes dentro do passo a frente é descrito pelo analista no início do processo de busca. É importante ressaltar que as abelhas podem visitar apenas um único componente para cada passo a frente. Na Figura 1 é apresentada uma execução do BCO na qual as abelhas realizam o terceiro passo a frente, onde *B* representa o número de abelhas e *N* o número de componentes.

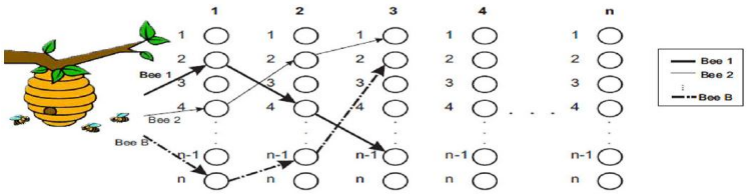

Figura 1: Execução do terceiro passo a frente [Lucic et Al 2006].

Obtendo novas soluções parciais, as abelhas retornam à colmeia, iniciando o passo a trás. Nesta etapa todas as abelhas compartilham a qualidade de suas soluções e comparam-nas com todas as soluções anunciadas. As abelhas então passam por um processo de decisão onde cada uma decide, com certa probabilidade, se continua fiel a sua solução (tornando-se uma recruta), ou abandona-a, tornando-se uma abelha não confirmada. As abelhas não confirmadas são obrigadas a selecionar uma das soluções anunciadas e seguir a abelha recruta correspondente.

No exemplo da Figura 1 as abelhas Bee1, Bee2, BeeB participaram do processo de decisão. Depois da comparação das soluções parciais a abelha BeeB decide por abandonar o caminho gerado e junta-se à abelha Bee2, como é mostrada na Figura 2 [Orco e Teodorovic 2008].

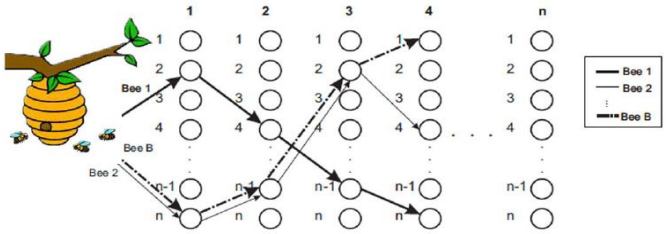

Figura 2: Situação do algoritmo no quarto passo [Lucic et Al 2006].

O BCO executa iterativamente até que uma das condições de parada seja satisfeita: o número de iterações chega ao limite, o número máximo de passos a frente/passos a trás chega ao limite, ou então a quantidade de iterações sem melhoria no resultado geral é atingida.

4.1 Trabalhos correlatos

Suguna e Thanushkodi (2010) desenvolveram um método híbrido entre Rough Sets e BCO para a seleção de atributos em bases de dados médicas do repositório UCI. Os autores utilizaram cinco bases de dados (Dermatology, Cleveland Heart, HIV, Lung Cancer Wisconsin) com número de atributos variando entre 9 e 56 e número de instâncias variando entre 32 e 699. Neste experimento algoritmo das abelhas conseguiu reduzir significativamente o número de atributos em todas as bases, resultando em bases com números de atributos entre 4 e 13. O número de abelhas utilizado foi 10 e a quantidade de iterações foi limitada em 1000.

Além disso, Forsati e Moayedikia (2012) também desenvolveram um trabalho similar, aplicando o BCO em bases do repositório UCI (Iris, Heart, Breast, Glass, Vowel e Vehicle), com número de atributos variando entre 4 e 19. O número de abelhas utilizado foi 20 e a quantidade de iterações foi fixada em 80. Os autores apresentaram os resultados do classificador com as bases originais e com os atributos reduzidos, as taxas de acerto foram similares, demonstrando um bom desempenho do BCO.

4.2 Algoritmo BCO aplicado ao problema de seleção de atributos

A aplicação do BCO tem o objetivo de selecionar os atributos mais relevantes da base, obtendo assim, um melhor desempenho na taxa de classificação. Primeiramente, é feito a definição dos parâmetros, que são: número de abelhas, número de iterações e o número de passo a frente (*NC*).

Nas execuções utilizadas neste trabalho o número de passos a frente foi definido de modo que a abelha interrompe o processo quando não existe mais melhoria na solução, desta forma o número de atributos selecionados é variável. No início do processo de busca, todas as abelhas estão na colmeia. O Algoritmo 1 apresenta o pseudo-código do BCO utilizado neste trabalho:

Algoritmo 1: Pseudo código do BCO.
1 **início** 2 uma solução vazia é atribuída para cada abelha

```
3   para cada abelha faça:
4       k recebe 1 // (contagem de movimentos construtivos no passo a frente)
5       Avalia todos os possíveis movimentos construtivos;
6       Escolhe um movimento usando o método da roleta;
7       k = k + 1;
8       se k ≤ NC então:
9           Volte a etapa 5;
10  Todas as Abelhas voltam para a colmeia; // (Iniciando o passo a trás)
11  Avalia a solução parcial de cada abelha;
12  Cada abelha decide se continua com sua exploração;
13  Para cada seguidora, escolhe uma nova solução das recrutas através de uma roleta;
14  se solução não completa então:
15      Volte a etapa 3;
16  Avalia todas as soluções e escolhe a melhor;
17  se Critério de parada não for satisfeito então:
18      Volte a etapa 3;
19  Mostrar a melhor solução encontrada;
20 fim
```

Durante o passo a frente, as abelhas voam no espaço de busca para que um certo número de atributos sejam visitados, para que isso aconteça foi construída uma roleta com base na métrica *chi-square*. As abelhas então retornam a colmeia para trocar as informações sobre a qualidade de suas soluções parciais geradas (etapas de 3 a 11), a medida de qualidade é baseada na métrica DEC.

Na colmeia as abelhas passam por um processo de decisão de lealdade, ou seja, comparam todas as soluções geradas e decidem com uma certa probabilidade se vão continuar leais às soluções atuais, o calculo é definido conforme apresentado na Equação 4:

$$P_b^{u+1} = e^{-\frac{O_{max}-O_b}{u}}$$

Equação 4: Fórmula para o cálculo de lealdade das abelhas.

Onde *Ob* é o valor atual da solução parcial da abelha em questão, e *Omax* é a melhor solução parcial de todas as gerações e *u* refere-se ao número de passos a frente.

A partir do momento que as abelhas abandonam suas soluções parciais, elas tornam-se abelhas não confirmadas, e com uma certa probabilidade, decidem por seguir uma das recrutas. A probabilidade de uma abelha seguir uma recruta é definida conforme apresentado na Equação 5:

$$P_b = \frac{O_b}{\sum_{k=1}^{R} O_k}$$

Equação 5: Fórmula para o cálculo da probabilidade de seguir outra solução.

Onde Ok representa o valor normalizado da função objetivo do k-th função parcial anunciada, e R denota o número de recrutas.

- **. Testes e Resultados**

Os testes do algoritmo foram realizados baseados em 12 bases de dados numéricas disponíveis no repositório UCI Machine e Learning Repository, as bases utilizadas foram: breast, bupa, ecoli, glass, haberman, heart, ionosphere, lettera, new thyroid, pima, set-image e vehicle, cada uma delas com atributos e instâncias distintas, na Tabela 1 são apresentados o total de atributos e instâncias contidas em cada base. Nenhuma das instâncias utilizadas durante o treinamento foi utilizada na etapa de testes.

Tabela 1: Bases de dados.

Base	Nº de Atributos	Nº Instâncias
Breast	9	3415
Bupa	6	1725
Ecoli	7	1680
Glass	9	1070
Haberman	3	1530
Heart	13	1350
Ionosphere	33	1755
Lettera	16	99995
New-Thyroid	5	1075
Pima	8	3840
Sat-Image	36	32175

A etapa de testes foi dividida em duas partes: a calibração dos parâmetros (localização da melhor combinação) e a execução do algoritmo em todas as bases com os parâmetros selecionados. Os parâmetros utilizados na etapa de calibração foram a quantidade de abelhas, variando entre 3, 5, 10, e a quantidade de iterações do algoritmo, variando entre 3, 5. O classificador utilizado foi o J48, e os resultados foram baseados na média de 10 execuções.

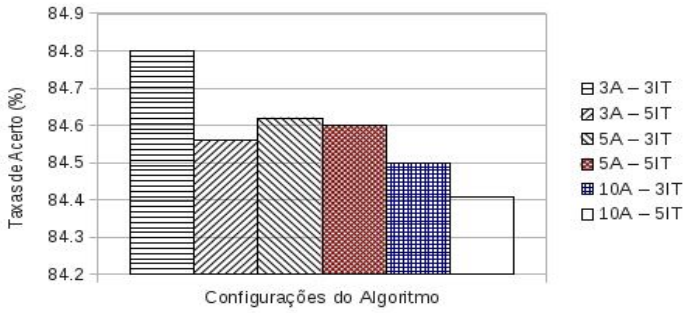

Figura 3: Taxa de acerto com diversas configurações de parâmetros (etapa de treinamento).

São apresentados na Figura 3, o gráfico com a média da taxa de acerto do classificador das 10 execuções de cada parâmetro para todas as bases de dados. Os parâmetros estão

representados no gráfico pelas siglas A e IT, sendo respectivamente a quantidade de abelhas e o número de iterações. É possível observar que a utilização de poucas abelhas e poucas iterações (no caso 3 abelhas e 3 iterações) implicou em uma taxa de acerto do classificador relativamente boa em relação as configurações que exigem mais processamento (como por exemplo 10 abelhas e 5 iterações). Isto se dá pelo fato de que muitas iterações geram um maior número de atributos, assemelhando-se à base original, o que não é o objetivo da seleção de atributos pois muitos atributos pioram o resultado do classificador.

Após a análise da calibração dos parâmetros a combinação escolhida para os testes e comparações foi 3 abelhas e 3 iterações, os resultados obtidos foram satisfatórios e o custo computacional se mostrou baixo. A Figura 4 apresenta um comparativo das taxas de acerto do classificador com as bases originais e as bases filtradas (0,5 para 50% e 1 para 100%). É notável que as bases filtradas resultaram em melhor taxa de acerto pois os atributos inúteis e/ou pouco influentes foram removidos. A quantidade de atributos selecionados foi, em média, de 3 a 4 atributos.

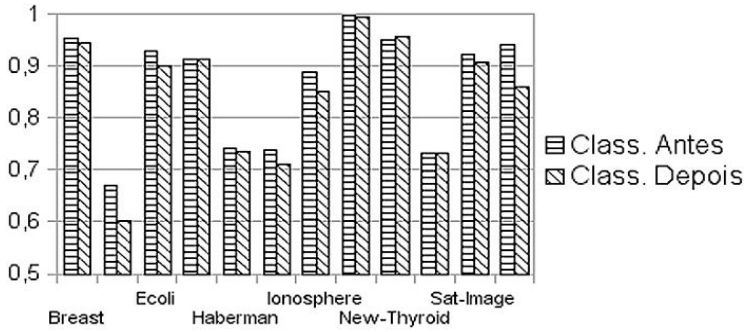

Figura 4: Comparativo entre as bases originais e as bases filtradas.

O critério principal para avaliar a carga computacional consumida pelo algoritmo foi a quantidade de avaliações da função objetivo necessárias em cada execução, sendo assim as execuções com menos iterações são as que exigem menos carga computacional.

- **. Considerações Finais**

Classificar os dados com um grande número de dados pode ser um processo difícil, o treinamento desses dados pode ser uma tarefa demorada que não produz resultados consistentes. Este trabalho apresentou um método de reduzir o número de atributos, selecionando apenas os mais relevantes sem perder informações importantes da base.

O algoritmo BCO aplicado ao problema de seleção de atributos se portou como esperado, selecionou subconjuntos de atributos os quais proporcionaram resultados satisfatórios. Foi possível perceber que o classificador J48 trabalha melhor, ou seja, obtém em uma melhor taxa de acerto, quando os atributos irrelevantes são removidos da base de dados. Além disso, é notável, através dos resultados do classificador, que configurações do algoritmo que geraram muitos atributos (como 10 abelhas e 5 iterações) resultaram em classificações piores do que os que geraram menos atributos (como 3 abelhas e 3 iterações).

Também é importante ressaltar que a configuração do o algoritmo que utilizou 3 abelhas e 3 iterações, apresentou um baixo custo computacional e bons resultados de precisão com relação às outras configurações. Isto incentiva aplicá-la a problemas complexos e que tradicionalmente demandam alto custo computacional.

- **. Referências**

Armentano, M. A. A. "Uma Metodologia na Utilização de Algoritmos Genéticos na Seleção de Atributos em Mineração de Dados." Monografia, 2005.

Castanheira, L.G. "Aplicação de Técnicas de Mineração de Dados em Problemas de Classificação de Padrões". Dissertação de mestrado, UFMG, 2008.

Lee, H. D. "Seleção de atributos importantes para a extração de conhecimento de bases de dados". Tese de Doutorado, Universidade de São Paulo, 2005.

Lucic, D. T, Markovic G. and Orco M. D. "Bee Colony Optimization: Principles and Applications". Setembro 2006.

Orco, M. D. and D. Teodorovic. "Mitigating Traffic Congestion: Solving the Ride-Matching Problem by Bee Colony Optimization". Janeiro 2008.

Pereira, R. B. "Seleção Lazy De Atributos para a Tarefa de Classificação." Dissertação de Mestrado, Universidade Federal Fluminense, 2009.

Pham, D. T., A. Ghanbarzadeh, E. Koc, S. Otri, S. Rahim and M. Zaid. "The Bees Algorithm A Novel Tool for Complex Optimisation Problems". 2006.

Pila, A. D. "Seleção de Atributos Relevantes para Aprendizado de Máquina Utilizando a Abordagem de Rough Sets". Dissertação de Mestrado, USP, Abril 2005.

Soares, M. V. S. "Avaliação de uma Abordagem Lazy de Seleção de Atributos Baseada na Medida de Consistência". Monografia, 2010.

Spolaôr, N. "Aplicação de Algoritmos Genéticos Multiobjetivo ao Problema de Seleção de Atributos." Dissertação de Mestrado, Universidade Federal do ABC, 2010.

Teodorovic and M. D. Orco. "Bee Colony Optimization: A Cooperative Learning Approach to complex Transportation Problems". Setembro 2005.

Teodorovic, D, T. Davidovic and M. Selmic. "Bee Colony Optimization: The Applications Survey". Fevereiro 2010.

Forsati, R., Moayedikia, A., Keikha, A., and Shamsfard, M. "A Novel Approach for Feature Selection based on the Bee Colony Optimization". International Journal of Computer Applications, 43(8):13–16.

Suguna, N. and Thanushkodi, K. "A Novel Rough Set Reduct Algorithm for Medical Domain Based on Bee Colony Optimization". arXiv preprint arXiv:1006.4540.