

Otimização com Colônias de Formigas: Aplicações

Gisele L. Pappa

Otimização por Colônia de Formigas

- Método de busca onde um conjunto de agentes (formigas) constroem incrementalmente soluções candidatas para um problema
- Contrução é probabilística e guiada por:
 1. Informações heurísticas sobre o problema
 2. Por um memória compartilhada por todos os agentes
 - Trilha artificial de feromônio

Otimização por Colônia de Formigas

- Cada **caminho** representa uma possível solução
- Quantidade de feromônio depositada é proporcional a **qualidade** da solução
- Se formigas têm que escolher entre dois caminhos, a probabilidade delas seguirem o caminho melhor é maior

Aplicações

- Caixeiro viajante
 - Utilizada por empresas de telefonia para rotas de tráfico de ligações telefônicas
- Robótica coletiva
- Mineração de dados



Ant-Miner

Paper original: [Parpinelli et al 2002]

Várias variações: <https://github.com/febo/myra>

Classificação de dados

- Classificação: atribuir a cada exemplo de um conjunto de treinamento uma classe, onde o conjunto de classes é finito e pré-definido.
- Existem diversos tipos de modelos de classificação
 - Árvores de decisão, redes neurais, modelos de regras

Indução de regras de classificação

- Uma regra segue o formato

IF $\langle L \rangle$ then $\langle \text{classe} \rangle$

onde

L é uma lista de condições do tipo $\langle \text{atributo operador valor} \rangle$

classe pertence ao conjunto de k valores de classes $\{c_1, c_2, \dots, c_k\}$

Definição do Problema

- Dado: um conjunto de dados de treinamento
- Objetivo: gerar regras de classificação que representem esse conjunto
- Saída: lista ordenada de regras de classificação

Conjunto de treinamento

Classe

Outlook	Temp	Humidity	Windy	Play
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Rainy	Mild	High	True	No

Regras de decisão

Exemplo de regra:

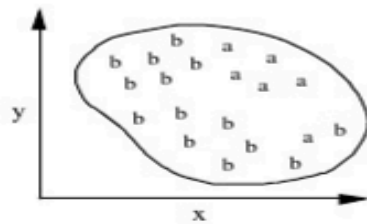
IF (humidity = high) and (outlook = sunny) THEN
play = no (3.0/0.0)

Exemplo de regra

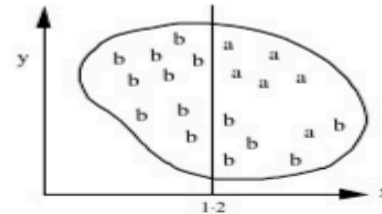
Outlook	Temp	Humidity	Windy	Play
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Rainy	Mild	High	True	No

IF (humidity = high) and (outlook = sunny) THEN play=no (3.0/0.0)

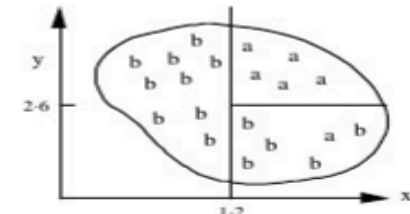
Exemplo de Regras



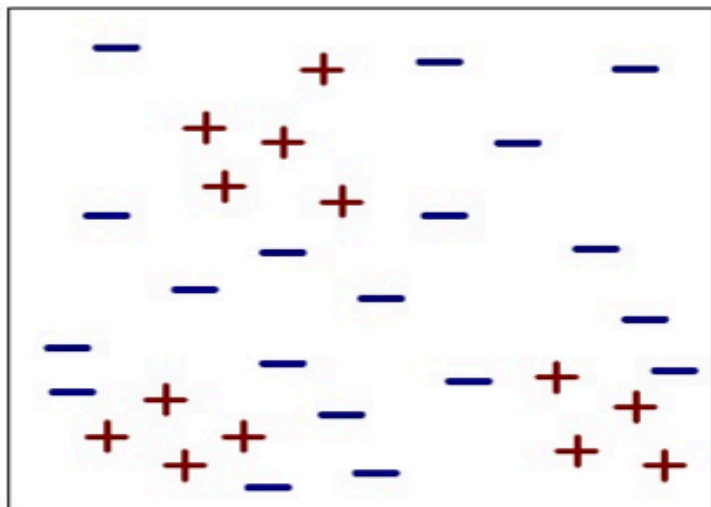
If true then class = a



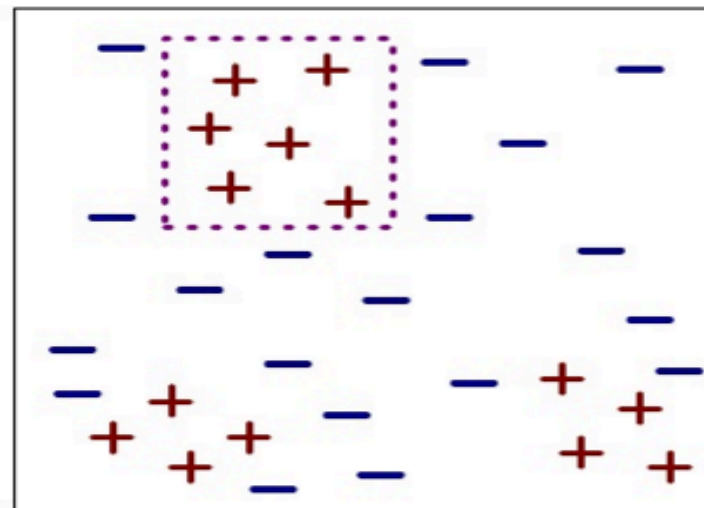
If $x > 1.2$ then class = a



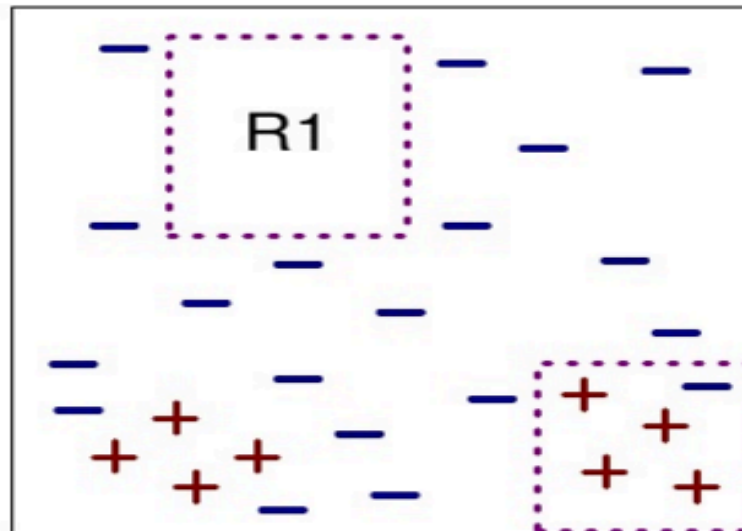
If $x > 1.2$ and $y > 2.6$ then class = a



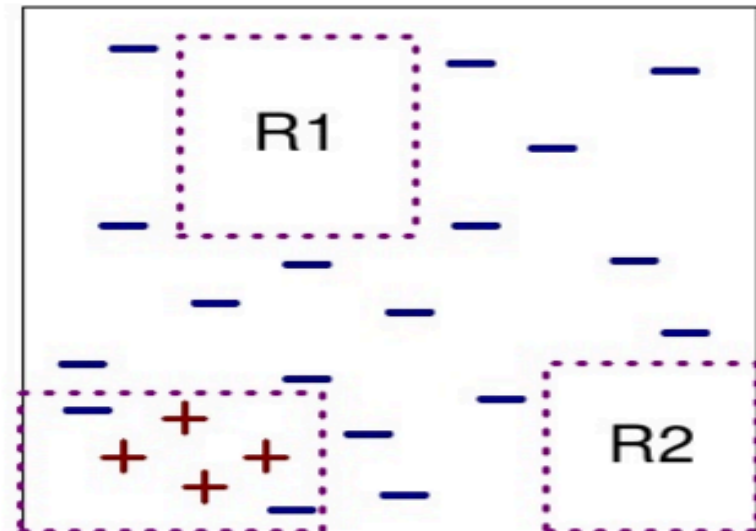
Dados Originais



Passo 1



Passo 2



Passo 3

Sequential Covering

while existirem elementos no conjunto de treinamento

 Aprenda uma regra que cubra parte dos exemplos de
 treinamento

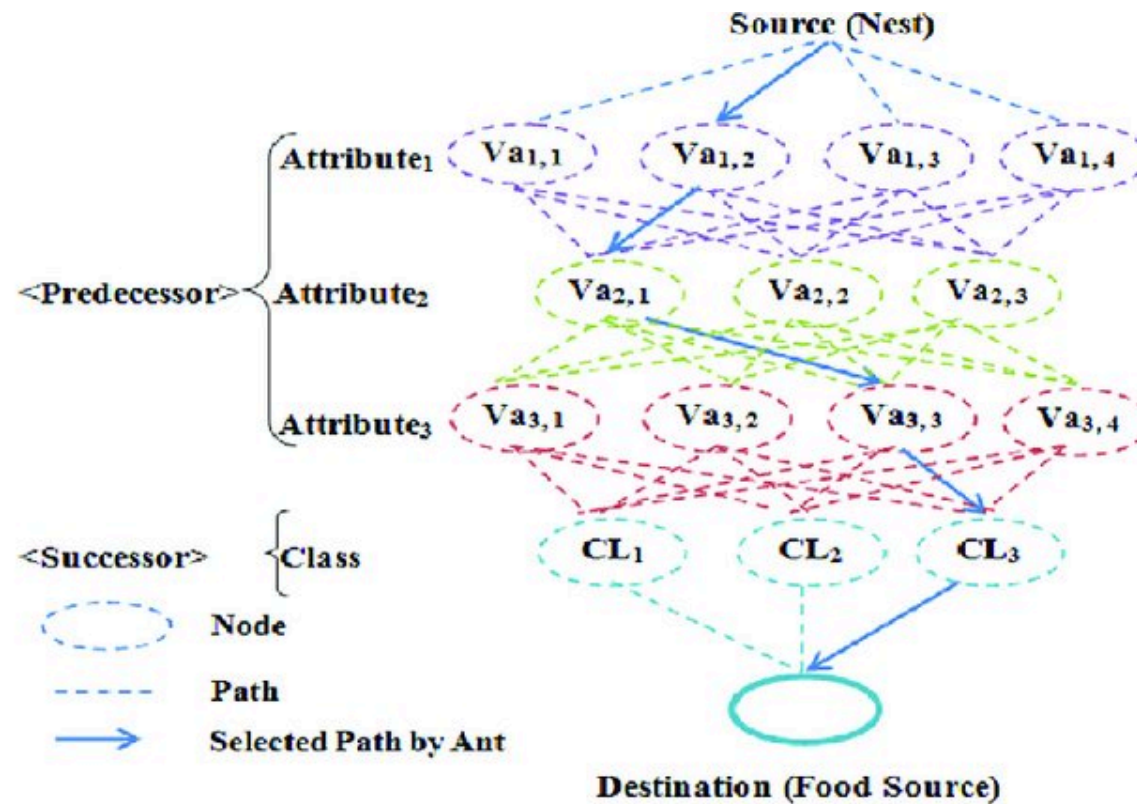
 Remova os exemplos cobertos pela regra do conjunto de
 treinamento

end while

Ant-Miner

- Segue a abordagem de cobertura sequencial
 - Formigas trabalham para gerar uma regra de decisão
 - ACO executado várias vezes (a cada execução um número menor de exemplos é considerado)
- Cada formiga inicialmente representa uma “regra vazia”
- Formigas incrementalmente adicionam um **termo (condição) a regra (par atributo+valor)** de acordo com uma métrica de entropia e a quantidade de feromônio depositada

Ant Miner



Ant-Miner

- Cada regra deve ter uma cobertura mínima
 - Serve de critério de parada e controla *overfitting*
- Ao final da iteração, a classe da regra é escolhida de acordo com a classe da maioria dos exemplos cobertos.
- Versão inicial só trabalhava com atributos categóricos

Ant-Miner

```
WHILE (No. of cases in the Training set > Max-uncovered allowed)
  i=0;
  REPEAT
    i=i+1
    Anti incrementally constructs a classification rule
    Prune the just constructed rule
    Update the pheromone of the trail followed by Anti
  UNTIL (i ≥ No-of-Ants) or (Anti covers less examples than a predefined threshold)
  Select the best rule among all constructed rules
  Remove the cases correctly covered by the selected rule from the training set
END WHILE
```

AntMiner

Inicialização do feromônio

$$\tau_{ij}(t=0) = \frac{1}{\sum_{i=1}^a b_i}$$

a é o número de atributos,
b_i é o número de valores q a pode assumir

Regra de transição

$$P_{ij} = \frac{\eta_{ij}^{\alpha} \tau_{ij}^{\beta}}{\sum_{i=1}^a \sum_{j=1}^{b_i} \eta_{ij}^{\alpha} \tau_{ij}^{\beta}}, \forall i \in I$$

η_{ij} – quanto maior seu valor, mais importante para o classificação é o valor j do atributo i

Ant-Miner

Avaliação: Qualidade da regra é dada de acordo com a entropia normalizada, que varia de 0 a $\log_2 k$

$$\eta_{ij} = \frac{\log_2 k - H(W | A_i = V_{ij})}{\sum_{i=1}^a \sum_{j=1}^{b_i} (\log_2 k - H(W | A_i = V_{ij}))} \quad \begin{array}{l} k \text{ é o número de classes,} \\ W \text{ é a classe} \end{array}$$

$$H(W | A_i = V_{ij}) = - \sum_{w=1}^k (P(w | A_i = V_{ij}) \cdot \log_2 (w | A_i = V_{ij}))$$

Existe também um método guloso para podar as regras

Ant-Miner- Resultados

Data Set	Ant-Miner's predictive accuracy (%)	CN2's Predictive accuracy (%)
Ljubljana breast cancer	75.28 ± 2.24	67.69 ± 3.59
Wisconsin breast cancer	96.04 ± 0.93	94.88 ± 0.88
tic-tac-toe	73.04 ± 2.53	97.38 ± 0.52
dermatology	94.29 ± 1.20	90.38 ± 1.66
hepatitis	90.00 ± 3.11	90.00 ± 2.50
Cleveland heart disease	59.67 ± 2.50	57.48 ± 1.78

Data Set	No. of rules		No. of terms / No. of rules	
	Ant-Miner	CN2	Ant-Miner	CN2
Ljubljana breast cancer	7.10 ± 0.31	55.40 ± 2.07	1.28	2.21
Wisconsin breast cancer	6.20 ± 0.25	18.60 ± 0.45	1.97	2.39
tic-tac-toe	8.50 ± 0.62	39.70 ± 2.52	1.18	2.90
dermatology	7.30 ± 0.15	18.50 ± 0.47	3.16	2.47
hepatitis	3.40 ± 0.16	7.20 ± 0.25	2.41	1.58
Cleveland heart disease	9.50 ± 0.92	42.40 ± 0.71	1.71	2.79

Bibliografia

- M. Dorigo, G. Di Caro and L. M. Gambardella. Ant Algorithms for Discrete Optimization. *Artificial Life*, 5(2):137-172.
- Parpinelli, R.S., Lopes, H.S., Freitas, A.A. "Data mining with an ant colony optimization algorithm". *IEEE Transactions on Evolutionary Computation*, special issue on Ant Colony Algorithms, v. 6, n. 4, p. 321-332, August, 2002