

Computação Evolucionária

Prof. Heitor Silvério Lopes

hslopes@utfpr.edu.br
<http://silverio.net.br/heitor>



Programação Genética

✦ Parte I - Fundamentos:

- Origens, literatura, contextualização, resolução de problemas, indução de programas
- Conjuntos de funções e terminais, geração da população inicial, medidas de *fitness*, operadores principais e secundários, critério de parada

✦ Parte IIa - Aplicações:

- Exemplos de aplicação clássicas (cap. 7):
 - Regressão simbólica
 - Formiga artificial

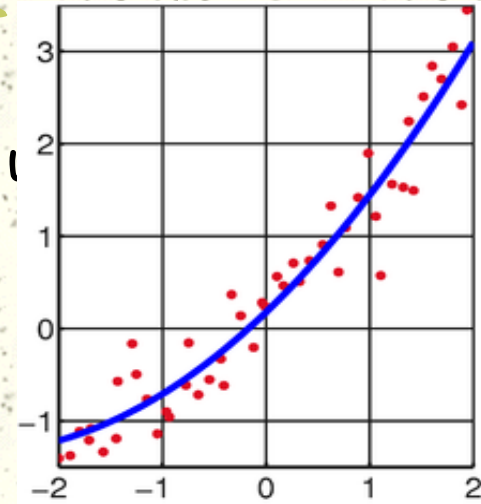
✦ Parte IIb - Aplicações:

- Evolução de comportamentos emergentes (cap. 12):
 - Busca de alimento
 - Priorização de tarefas
- Evolução da classificação (cap. 17):
 - Mineração de dados
 - Reconhecimento de padrões
- Evolução de estratégias (cap. 15):
 - Estratégia de jogo perseguidor-evasor
- Software Lil-GP

Exemplo clássico de aplicação #1: Regressão simbólica (Cap. 7)

* Regressão numérica:

- Ajuste de uma função multidimensional a um conjunto de pontos experimentais
 - Linear: $y(x)=a.x+b$
 - Quadrática: $y(x)=a.x^2+b.x+c$
 - Potência: $y(x)=a.x^b$
 - Outras: exponencial, logaritmica, polinomial
- O objetivo é encontrar o conjunto de coeficientes da equação (a,b,c...) que minimize uma medida de erro entre os valores dados e os computados.
- Há um método convencional para resolver este problema, p.ex. Método dos Mínimos Quadrados



Regressão simbólica

Regressão simbólica:

- Não apenas os coeficientes da equação, mas também a própria função a ser ajustada são desconhecidos.



Aplicabilidade:

- Identificação de sistemas (Eng. de Controle)
- Construção de modelos analíticos com dados experimentais
- *Data mining*
- Modelagem e previsão de séries temporais



Modelagem de Problemas de Regressão simbólica

Conjunto de terminais:

- Determinar as variáveis independentes (x_i) e a variável dependente (y), tal que $y=f(x_i)$

Conjunto de funções:

- É dependente do problema:
 - P.ex. funções aritméticas $\{+, -, *, \%$ e outras matemáticas baseadas no conhecimento do problema $\{\text{sen}, \text{cos}, \text{exp}, \text{rlog}, \text{psqrt}\}$

Regressão simbólica

Casos de *fitness*:

- O conjunto de dados de entrada, *t*-uplas ordenadas.

Função de fitness:

- Medida de **erro**: somatório (sobre os casos de *fitness*) das diferenças absolutas entre o valor da variável dependente produzida pela equação e o valor real da variável dependente.

$$fitness_{raw} = \sum_{i=1}^{N_c} |S(i) - C(i)|$$

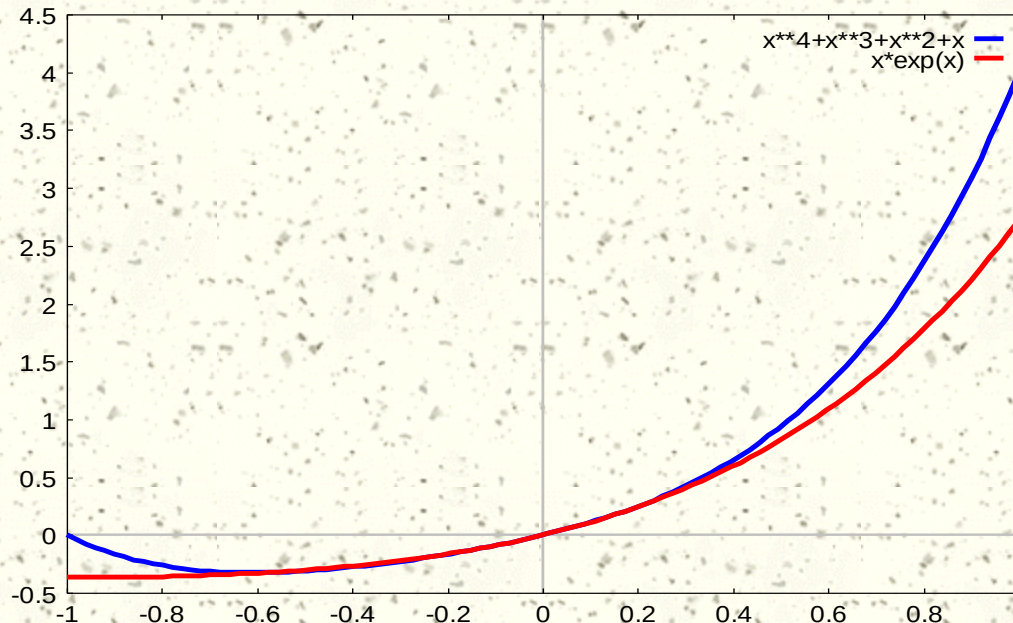
Exemplo: Regressão simbólica

Equação original: $y(x) = x^4 + x^3 + x^2 + x$

Melhor indivíduo na geração inicial:

```
(* x (+ (+ (- (% x x) (% x x)) (sin (- x x))) (rlog (exp (exp x))))))
```

Equação simplificada: $y = xe$

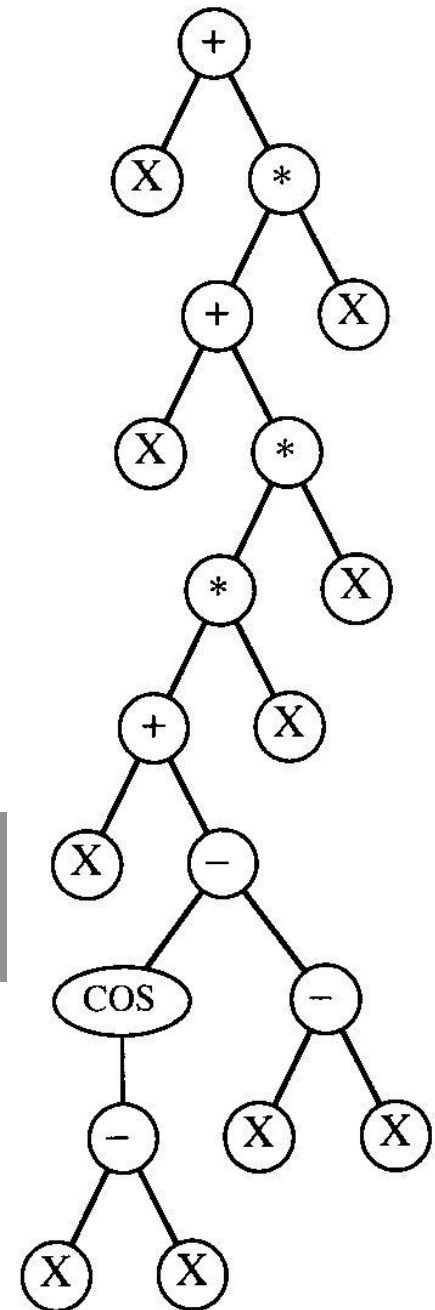


Regressão simbólica

- ✦ Melhor solução obtida para o problema encontrado após 37 gerações com população de 500 indivíduos (≈ 19000 avaliações):

✦ Expressão S:

$(+ x (* (+ x (* (* (+ x (- (\cos (-x x)) (- x x))) x) x)) x))$



Regressão simbólica no mundo real

- # Presença de ruído
 - # Múltiplas variáveis dependentes acopladas
 - # Ajuste fino de constantes
 - # Dependência temporal das variáveis independentes
 - # Descontinuidades
 - # Estocasticidade
-
- # Ainda assim, GP pode ser um bom ponto de partida !!



Exemplo prático #1: Previsão da vazão mensal de rios



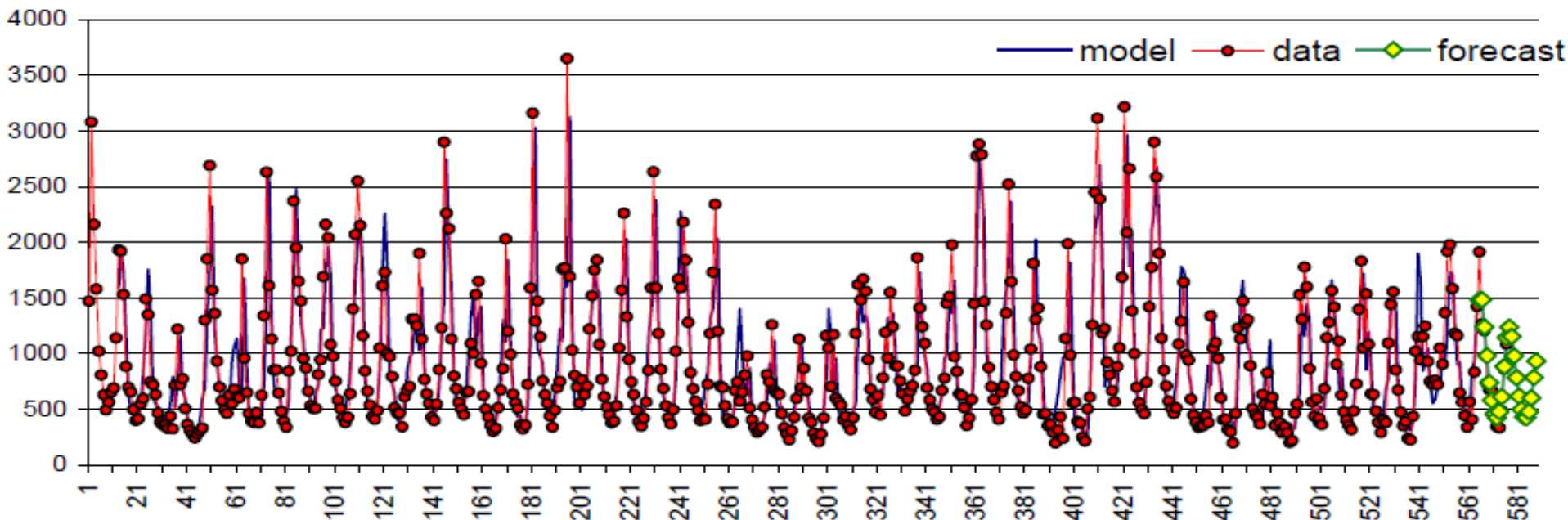
- ✦ Dados reais: vazão média mensal do rio Grande na barragem de Furnas, dados de 1931-1978
- ✦ Sazonalidade conhecida: observação dos 12 meses anteriores para prever o mês seguinte (*one-step ahead*)
- ✦ A melhor solução encontrada obteve um bom ajuste aos dados (NMSE=0,0853 e R=0,9301 no conjunto de teste)

$$\left(\frac{11.500}{(T4 / ((T11 * T10) / T4))} \right) + \left(\frac{(T1 + ((T11 - T9) / T4)) + 0.000}{((T1 - T9) / (1.360 - T3) / 7.353)} \right) + \left(\frac{11.500}{(T4 / ((T11 * T10) / T4))} \right) + \left(\frac{15.001}{(T1 / T3)} \right)$$

$$x_t = \left(\frac{23. \left(\frac{x_{t-11} \cdot x_{t-10}}{x_{t-4}} \right) + x_{t-1} \cdot x_{t-4} + x_{t-11} - x_{t-9}}{x_{t-4}} \right) - 0.1. (x_{t-1} - x_{t-9} + 1.36. x_{t-3}) + \left(\frac{15. x_{t-3}}{x_{t-1}} \right)$$

Exemplo prático #1:

Previsão da vazão mensal de rios



Mais detalhes:

Lopes, H.S., Weinert, W.R.

[A gene expression programming system for time-series modeling](#). In: **Proc. XXV Iberian-Latin American Congress on Computational Methods in Engineering**, 2004

Exemplo prático #2:

Identificação de sistemas

- # É um importante problema da Engenharia de Controle
- # Para controlar um sistema complexo é imprescindível ter o modelo matemático que descreve o funcionamento deste sistema
- # Quando nem o modelo e nem os parâmetros são conhecidos, não há método analítico para resolver
- # PG / PEG pode ser aplicado com sucesso



Exemplo prático #2: Identificação de sistemas

* PEG foi aplicada a três problemas importantes, de áreas distintas:

- Modelagem de um fenômeno ecológico: crescimento de algas no Lago Eirie (USA)
- Modelagem de um processo químico: tanque de mistura de produtos ácidos e alcalinos
- Modelagem de uma usina termoeletrica (França)



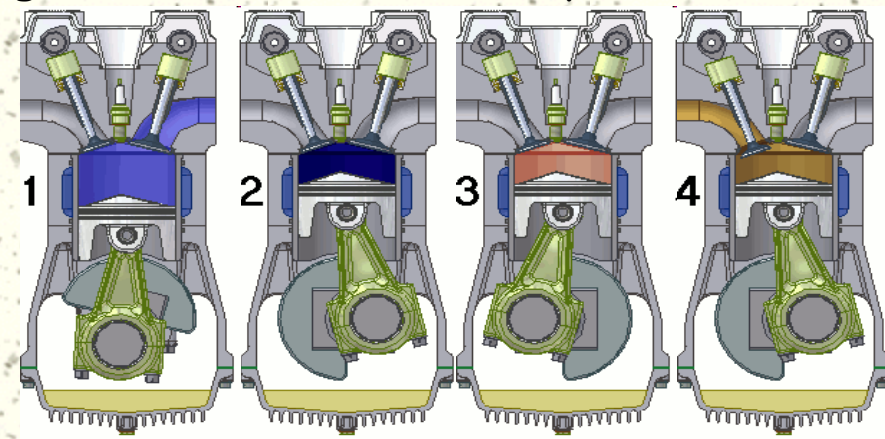
Lopes, H.S., Weinert, W.R. [EGIPSYS: a enhanced gene expression programming approach for symbolic regression problems](#). **International Journal of Applied Mathematics and Computer Science**, v. 14, n. 3, p. 375-384, 2004.

Exemplo prático #3: Identificação de sistemas

✦ Obtenção da função de transferência de um motor a ciclo Otto:

- O motor de ciclo Otto foi inventado por Nikolaus Otto em 1876, e posteriormente aperfeiçoado por Étienne Lenoir e Rudolf Diesel
- Variáveis de entrada (velocidade, torque, quantidade de combustível, ponto de ignição, taxa EGR) e saída (fumaça e gases NOx) foram obtidos experimentalmente em laboratório
- A função de transferência obtida foi suficientemente simples para ser compreendida e utilizada por engenheiros e reflete com precisão o funcionamento do motor

Guerra, E., Lopes, H.S., Godoy Jr., W.
Determinação da função de transferência de um motor ciclo Otto utilizando programação genética. In: **Anais do X Congresso Brasileiro de Inteligência Computacional**, p. 1-7, 2011



Exemplo clássico de aplicação #2: Formiga artificial (Cap. 7)



Cenário:

- Uma formiga-robô simulada tem as seguintes capacidades:
 - detectar comida diretamente à sua frente
 - virar para a esquerda ou direita
 - mover para frente.
- O "mundo" é uma "grade" com trilhas de comida, havendo alguns vazios nesta trilha.

Objetivo:

- Coletar o máximo de comida possível da trilha.
- Uma vez que há vazios na trilha, a formiga deve ter comportamento exploratório para quando não houver comida detectada, de modo a reencontrar a trilha de alimento.

Formiga artificial

- # Conjunto de terminais: $T = \{\text{MOVE, RIGHT, LEFT}\}$
- # Conjunto de funções: $F = \{\text{IF-FOOD-AHEAD, PROG2, PROG3}\}$
 - IF-FOOD-AHEAD (*,*) executa o primeiro argumento se há comida à frente e o segundo argumento se não há.
 - PROG2 (*,*) executa seu primeiro argumento e depois o segundo.
 - PROG3(*,*,*) executa seus três argumentos em sequência.
- # Exemplo:
 - move-se para frente se detectou comida, caso contrário vira para a esquerda:
 - (IF-FOOD-AHEAD (MOVE) (LEFT))

Formiga artificial

✦ *Fitness*:

- é dado como o número de pedaços de comida encontrados pela formiga num determinado número de passos temporais (*time-steps*).
- O programa é executado até atingir o número de passos pré-determinado.

Formiga artificial

- # Melhor solução para uma trilha em particular, encontrada após 21 gerações, com uma população de 500 indivíduos (≈ 11000 avaliações):

```
(IF-FOOD-AHEAD (MOVE)
  (PROG3 (LEFT)
    (PROG2 (IF-FOOD-AHEAD (MOVE)
      (RIGHT))
      (PROG2 (RIGHT
        (PROG2 (LEFT)
          (RIGHT))))
      (PROG2 (IF-FOOD-AHEAD (MOVE)
        (LEFT))
        (MOVE))))
```

Formiga artificial

