

# Computação Natural - TP1 - Programação Genética

Guilherme Torres (Departamento de Ciência da Computação - UFMG)

## 1 Introdução

O objetivo deste trabalho foi formular e escolher os melhores parâmetros para um algoritmo de aprendizado para regressão simbólica. Dada uma função  $f : \mathbb{R}^n \rightarrow \mathbb{R}$ , a regressão simbólica consiste em tentar aproximar essa função usando uma árvore constituída por terminais nas folhas (constantes ou variáveis) e operadores.

A heurística usada foi a programação genética, que usa de conceitos evolucionários para evoluir as soluções para o problema e obter indivíduos mais adaptados. É um algoritmo estocástico, ou seja, com vários componentes de aleatoriedade como geração inicial dos indivíduos e mutações, o que requer que muitos experimentos sejam realizados para tirar as conclusões sobre o trabalho, que serão discutidos nesse documento.

O algoritmo foi implementado em Python 3, na pasta `/src`. Instruções para executar o teste estão explicitadas no arquivo `README.md`. O programa também pode ser encontrado no repositório:

<https://github.com/torresguilherme/cn-tp1-gp>.

## 2 Implementação

### 2.1 Fundamentação

O funcionamento do algoritmo pode ser descrito, no plano geral, pelos seguintes passos:

1. Geração aleatória dos primeiros indivíduos;
2. Mostra a melhor, pior e média das fitness na população;
3. Adiciona o grupo elitista à nova população;

4. Repete até que o número de indivíduos na população possa ser reposto:
  - (a) Seleciona dois pais através do torneio;
  - (b) Aplica a probabilidade de cruzamento;
  - (c) Ao se obter um cruzamento, aplica a chance de mutação;
  - (d) Adiciona os filhos gerados à nova população;
5. Repete desde o passo 2, até que o número máximo de gerações seja alcançado.

Como é sugerido, há muitas variáveis envolvidas e deve-se encontrar o valor ótimo para elas por meio de experimentos.

## 2.2 Decisões de implementação

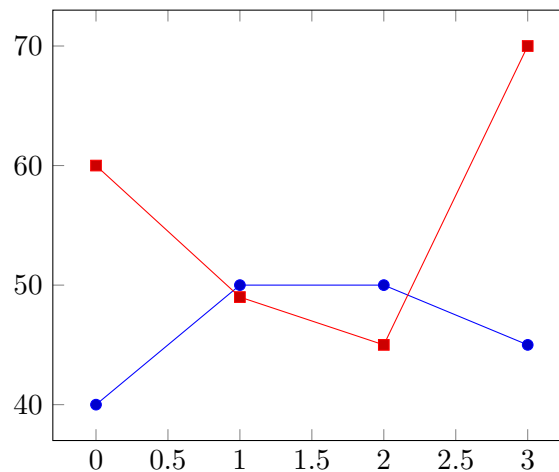


Figure 1: Data plot

### Indivíduo

Um indivíduo é representado por uma árvore, que tem um valor de uma função. Cada uma das folhas da árvore faz o papel de uma constante ou uma variável. As constantes são geradas aleatoriamente com números de ponto flutuante entre -10 e 10 (distribuição uniforme).

Os operadores escolhidos foram '+', '\*' e '/', respectivamente, para adição, multiplicação e divisão. Como a subtração é praticamente uma adição por um número negativo, não foi necessário adicionar um operador '-'.

## Geração aleatória de indivíduos

Cada indivíduo começa com um operador a priori, e sua árvore é estendida até que todos os caminhos acabem em um terminal. Como a profundidade máxima do indivíduo é 7, apenas terminais serão escolhidos aleatoriamente quando algum nó chegar na profundidade 6.

## Cálculo da Fitness

A fitness de um indivíduo é dada pelo seu erro quadrático em relação ao resultado dado para um conjunto de variáveis da função que se quer aprender. O erro de um indivíduo é obtido por

$$\sqrt{1/N \sum_{i=0}^{n-1} (Ev_i - y_i)^2}$$

sendo  $N$  o número de avaliações dadas no arquivo de teste,  $Ev_i$  a avaliação de um indivíduo para cada conjunto de variáveis,  $y_i$  o valor real dado no arquivo para aquele conjunto.

Como a fitness é uma medida de erro, quanto menor o erro, melhor é o indivíduo.

## Tamanho da população e número de gerações

O tamanho da população para um algoritmo de programação genética é tipicamente menor do que para um algoritmo genético, onde o objetivo é achar uma solução para um problema em vez de desenvolver um método para ele.

É razoável que a população varie, neste caso, entre 50 e 500. Por motivos de limitações em questão de tempo e recursos computacionais, o tamanho da população foi fixado em 50, assim como o número de gerações. Sabendo disso, é provável que seja necessária uma taxa de mutações e cruzamento maior para obter-se mais diversidade na população reduzida.

## Seleção natural

A seleção é realizada por torneio. Dado um tamanho de torneio  $T$ , uma amostra de tamanho  $T$  é selecionada também de forma aleatória entre os indivíduos e o vencedor é o que tiver a melhor fitness, no caso, o menor erro quadrático.

A variável  $T$  é o principal fator que define a intensidade da pressão seletiva na população. Quanto maior esse número é, mais difícil é a permanência de indivíduos com fitness pior do que a média na população. Por isso, o valor adequado para esse parâmetro também foi definido através de experimentos, dentro de um intervalo de valores.

## **Cruzamento**

O cruzamento também faz parte da seleção natural. Ele ocorre para definir a nova geração a ser avaliada no GP. Dados dois indivíduos que venceram seus respectivos torneios, a probabilidade de ocorrer um cruzamento entre eles subsequentemente é de  $p_c$  (entre 60% e 90%).

Uma das principais questões do cruzamento é a possibilidade de se manter a localidade ao cruzar duas árvores (se uma grande parte das árvores for cruzada, a variabilidade gerada pode levar a um indivíduo completamente diferente dos dois pais, o que aumenta drasticamente a aleatoriedade da busca).

Para evitar que um crossover provoque uma variedade maior do que deveria e destrua bons indivíduos, o modelo de crossover adotado é um método mais simples, porém ainda amplamente usado [1], que é a troca de duas folhas aleatoriamente escolhidas entre os pais.

## **Mutação**

Como o cruzamento é escolhido de forma bem conservadora, é papel da mutação provocar uma variabilidade mais drástica e promover a diversidade entre a população. A estratégia usada consiste em escolher um nó aleatório da árvore, podar os seus filhos e re-estender a árvore aleatoriamente a partir desse ponto [1].

Para cada indivíduo que surge de um cruzamento, a chance de ele sofrer uma mutação é  $p_m$  (entre 5% e 30%).

## **Elitismo**

Passar um pequeno grupo dos melhores indivíduos da população atual para a próxima geração sem submetê-los ao processo de seleção é um processo muito utilizado para fazer algoritmos convergir mais rápido e preservar bons genótipos. O elitismo é adotado neste projeto também.

Como o número de indivíduos da população é relativamente pequeno, julgou-se razoável manter o tamanho do grupo elitista em apenas um, para

que a pressão seletiva não fosse desproporcional e assim, evitando que todos os indivíduos ficassem iguais antes do número de iterações desejado.

### **3 Experimentos**

### **4 Conclusões**

### **5 Referências**

1. *A Field Guide To Genetic Programming*. - Riccardo Poli, William B. Langdon, and Nicholas F. McPhee, 2008