Regressão Simbólica - Trabalho Prático 1

Tiago Negrisoli de Oliveira

October 2018

1 Introduction

A regressão simbólica é uma forma de estimar uma função $f(x_1, ...x_n)$ que melhor descreve um conjundo de dados x_1,...,x_n. Para esse trabalho foi implementado um algorítmo baseado em programação genética para tentar encontrar essa função.

2 Modelo do Problema

Para a criação de um algorítmo baseado em programação genética, foram definidos alguns modelos de cada etapa, esses modelos são:

2.1 Indivíduo

Em uma regressão simbólica, a solução do problema é uma função $f(x_1, ..., x_n)$ que melhor descreve um conjunto de dados. Dessa forma, cada indivíduo da população é uma possível função. Para representar esses indivíduos, foi utilizado a estrutura de dados árvore binária, onde cada nó dessa árvore pode representar uma operação,uma variável ou uma constante. Um exemplo de indivíduo está a seguir:

O seguinte indivíduo representa a equação $(x_0 - 0,5) + (x_1 * x_1)$. Para ter $f(x_0, x_1)$ é preciso percorrer a árvore e substituir os valores das variáveis.

2.2 Geração da população inicial

Para a criação da população inicial, foi utilizado um método de geração de árvore aleatória, onde a probabilidade do nó ser de operação é $P(Op) = level_{nó}/Maxlevel$, onde $level_{nó}$ é o level do nó em relação a árvore, e o $Maxlevel$ é a profundidae máxima que a árvore pode ter, dada pelas configurações. Essa método constroi árvores com profundidades diversas, podendo parar em qualquer momento, visto que os nós de operação são não terminais e os de variável ou constante são terminais.
Além disso, para metade da população inicial, foi utilizado um método de $full$, onde a árvore será completa e terá a profundidade total sempre igual ao máximo

possível. Isso foi feito para garantir a diversidade maior da população e conseguir que as trocas de genes sejam melhores e diversas.

2.3 Fitness

A fitness utilizada para medir a qualidade de um indivíduo i perante o conjunto de dados é a distância normalizada do resultado da função representada por i e da $f(x_1, ..., x_n)$ dada pela entrada. A função utilizada é a seguinte:

2.4 Mutação

Para a mutação, existe apenas uma operação, que é substituir um nó da árvore por outra subárvore válida. Existe uma limitação nessa operação que é o tamanho gerado da sub-árvore, que nunca deve passar do limite máximo da árvore do indivíduo dada pelos parâmetros de configuração.

2.5 Cross-Over

O cross-over é uma operação que seleciona um nó aleatória de um indivíduo A e procura em um outro indivíduo B por uma sub-árvore válida que possa ser trocado com de A. A validez, assim como na mutação, é baseada apenas na altura da sub-árvore para evitar passar do limite dado. O cross-over possui a possibilidade de ser feita uma política elitista, em que um filho criado pelos indivíduos A e B, só será adicionado à população caso tenha uma fitness melhor que ambos os pais, caso contrário, será adicionado o melhor pai.

2.6 Seleção

A seleção utiliza torneio para escolher os indivíduos que irão realizar o cross-over ou mutação. O torneio funciona separando a população em k, os quais são formados aleatóriamente. Para cada um deles, escolhe-se o melhor indivíduo. Com os ganhadores são formados pares, e esses pares são passados para a mutação com probabilidade p ou para o cross-over com a probabilidade $1 - p$. Os indivíduos rejeitados passam pelo processo de torneio de novo até que toda a população tenha passado pelos operadores genéticos de mutação ou cross-over.

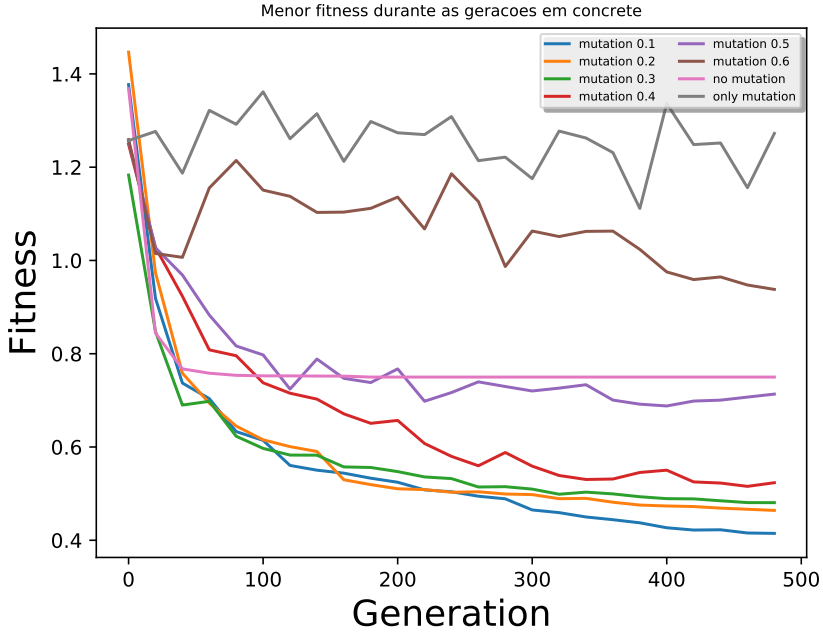
2.7 Condição de parada

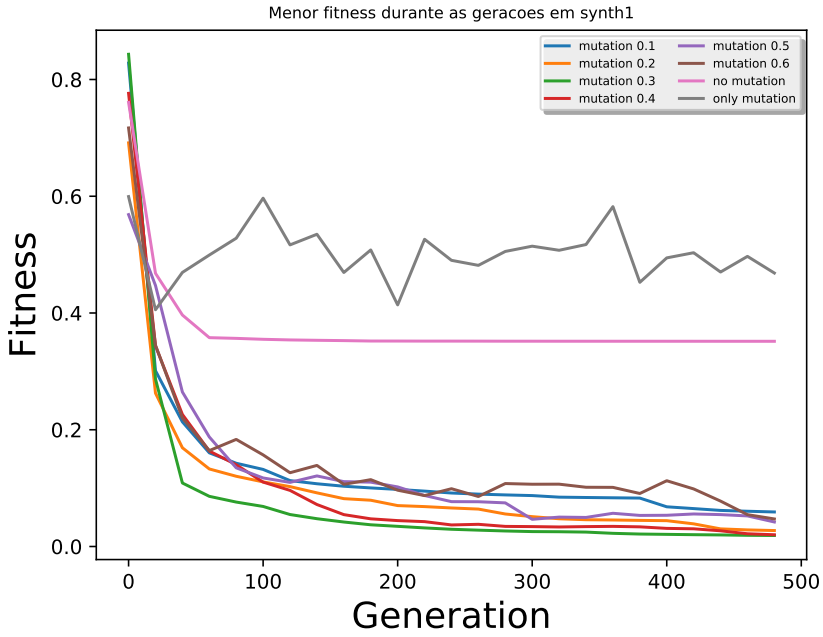
Para fins de experimentação, a condição de parada é dada apenas pelo número de gerações máxima. Essa escolha se da pela tempo que demoraria para alcançar uma solução com um erro muito baixo e devido ao grande número de testes feitos.

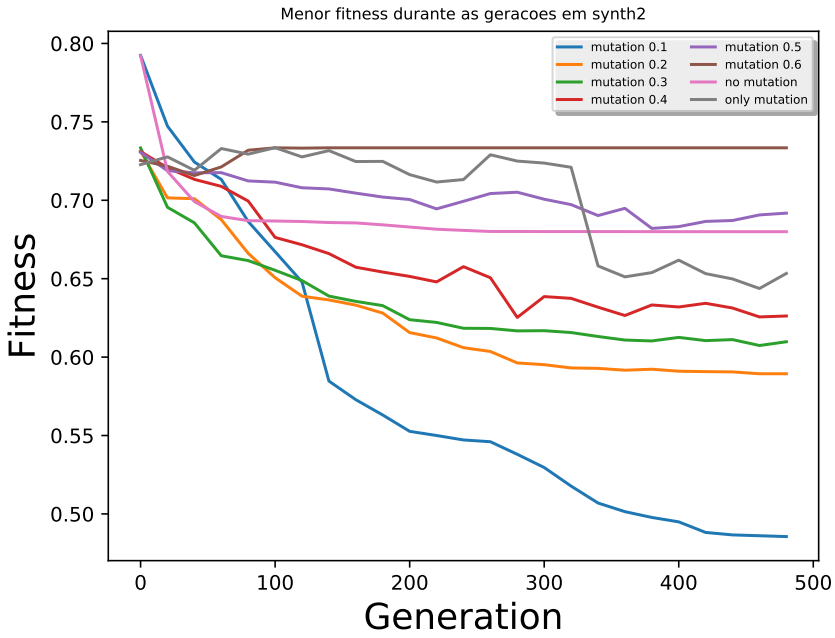
3 Análise experimental

Para testar o algorítmo e tentar encontrar um conjunto de parâmetros que sejam o melhor possível para cada conjunto de dados. Esses dados utilizados foram dois conjuntos de dados feitos de forma sintética e um que possuem valores experimentais reais.
Os experimentos foram realizados fixando os parâmetros de forma arbitrária e um deles variando para determinar qual valor ótimo dele.

3.1 Mutação


Menor fitness durante as geracoes em concrete
mutation 0.1
mutation 0.2
mutation 0.3
mutation 0.4
mutation 0.5
mutation 0.6
no mutation
only mutation
Fitness
Generation
0.4
0.6
0.8
1.0
1.2
1.4
0
100
200
300
400
500


Menor fitness durante as geracoes em synth1
mutation 0.1
mutation 0.2
mutation 0.3
mutation 0.4
mutation 0.5
mutation 0.6
no mutation
only mutation
Fitness
Generation
0.0
0.2
0.4
0.6
0.8
0
100
200
300
400
500

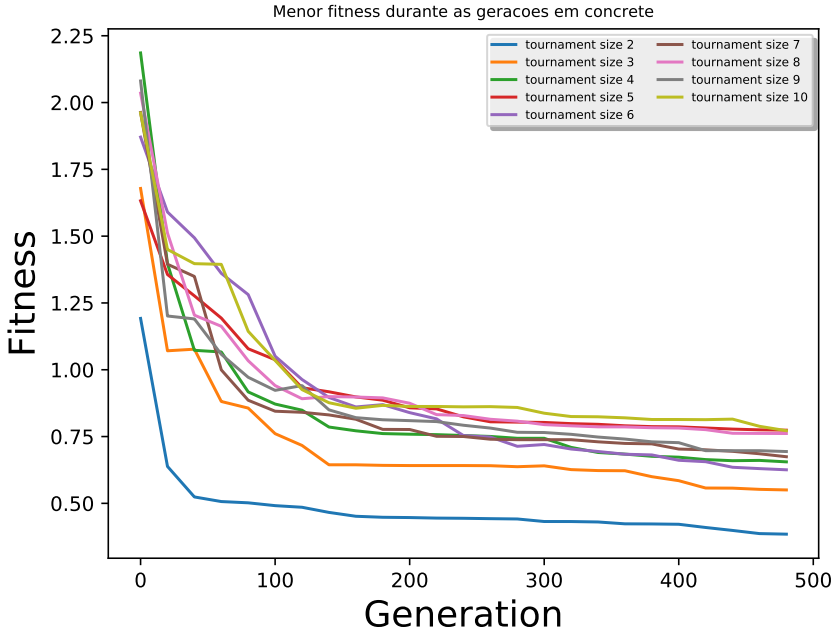


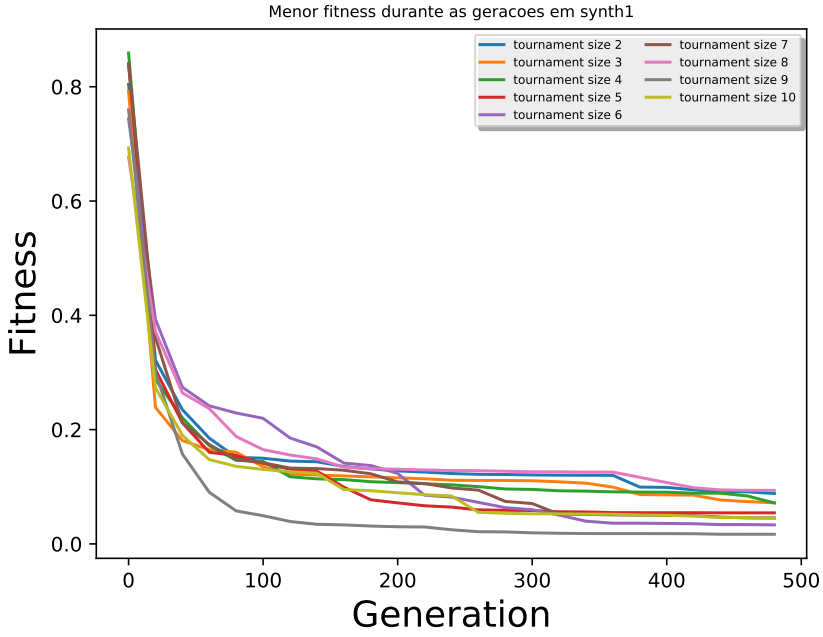
A figura acima mostra o valor da fitness do melhor indivíduo ao longo das 500 gerações em cada valor diferente de mutação. Ao analisar o grafico é possível ver que valores grandes de mutação, assim como nos caso de $1,0$ chance de ocorrência, a convergência não acontece, pois não existe um foco maior em um ponto no espaço de solução devido à prioridade pela exploração pelo mesmo.

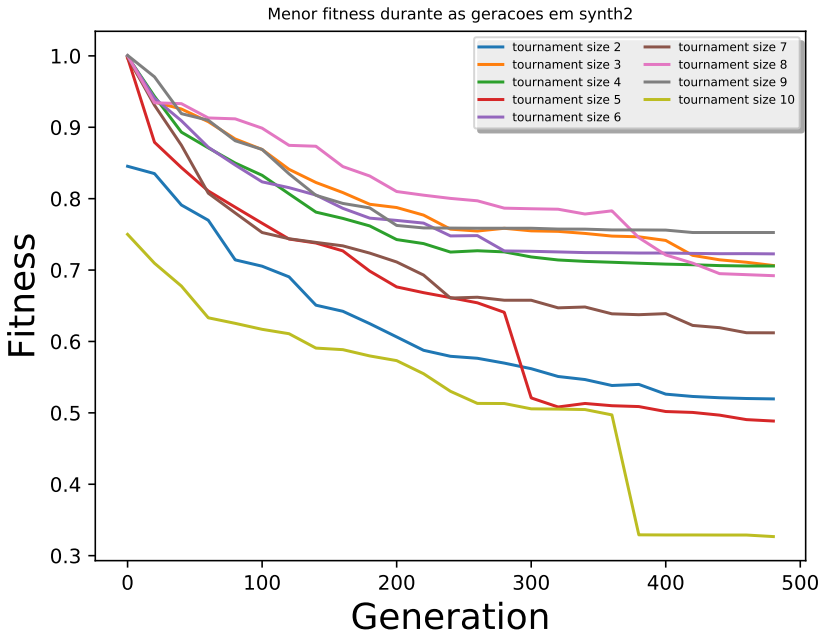
Em contrapartida, quando não existe mutação, a convergência ocorre muito rapidamente, porém, a solução se torna fixa e distante da ótima. Isso ocorre devido à não exploração do espaço de solução e ficar preso em um ótimo local.

Para valores pequenos de mutação, a melhor solução encontrada para a solução nos conjuntos de dados Synth2 e Concrete é 0.1 de chance de mutação, e em Synth1 é 0.3.

3.2 Tamanho do grupo do torneio

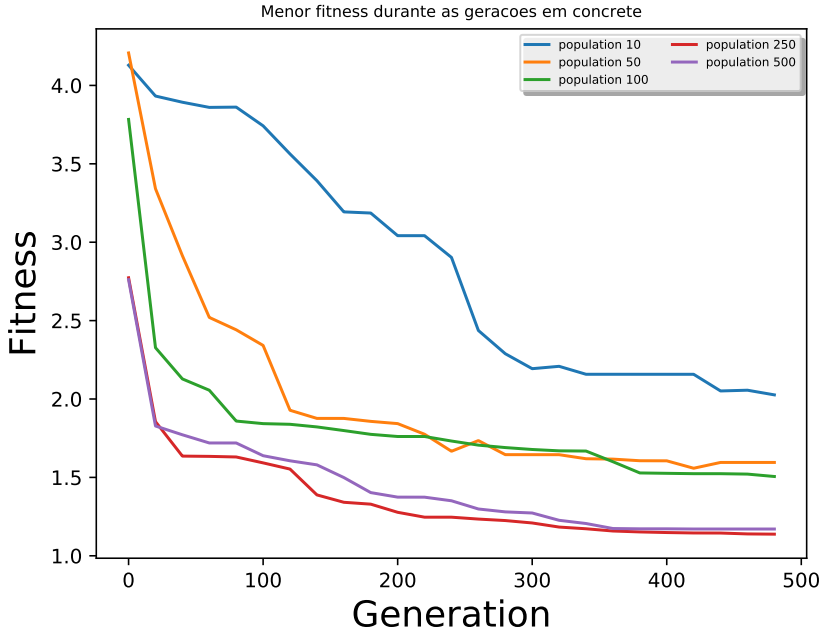


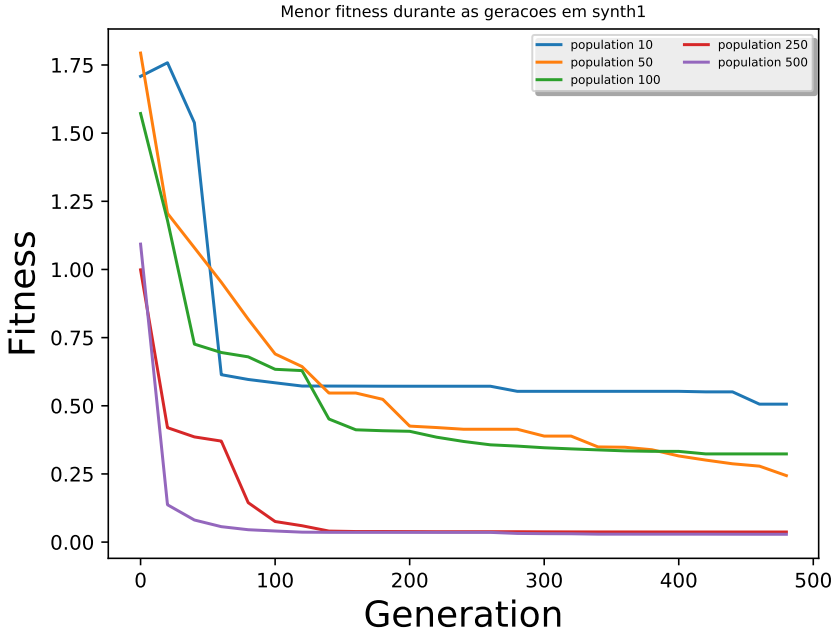


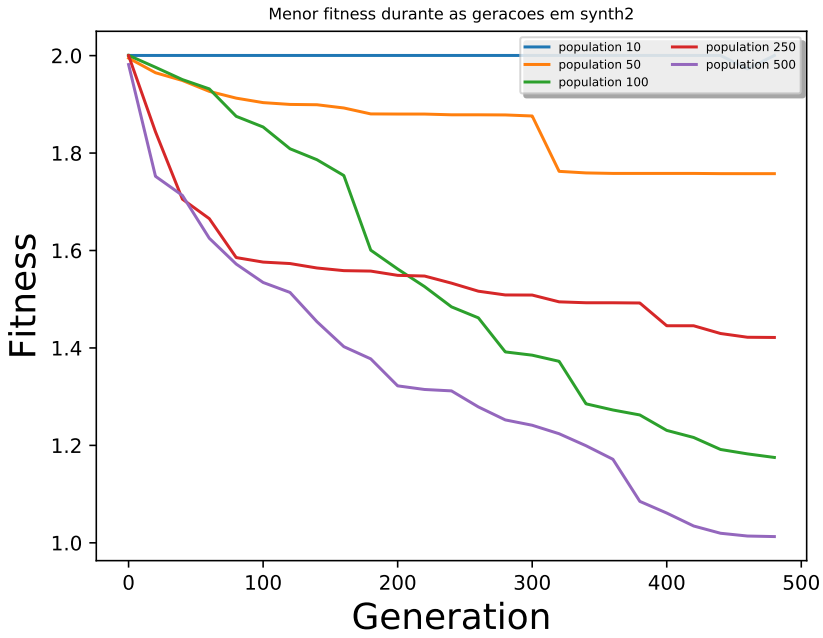


Podemos ver que o tamanho do torneio não tem um padrão em todos os testes, para o synth2 o melhor foi o 9, para o synth1 o 10 e para o concrete o 2. Uma outra análise possível é que para todo os valores do tamanho do grupo do torneio, em todos existe uma conversão que provavelmente não estagna em um ótimo local, logo qualquer valor seria válido e bom para achar uma solução.

3.3 Tamanho da população







Os graficos mostram a variação do número de indivíduos no processo genético do algorítmo. Dessa forma, podemos perceber que quanto maior a população mais rápida é a convergência e melhor o resultado esperado. Isso era esperado antes dos experimentos, pois quanto mais indívíduos, mais funções podem ser geradas e maior a distribuição no espaço de busca existe, consequentemente existe maiores chances da solução ótima e melhor ser encontrada.

4 Conclusões

Com esse trabalho, foi possível verificar e entender um pouco melhor o que cada parâmetro do algorítmo genético representa e impacta na solução. Além disso, este trabalho foi muito importante para conseguir compreender melhor como é o processo de experimentação e determinação desses parâmetros para tentar encontrar um conjunto de valores para cada um que melhor consegue gerar uma solução, de forma rápida e com melhor proximidade do ótimo.