September 30, 2018

**Abstract**

# 1 Introdução

Regressão simbólica é um tipo de análise de regressão que utiliza um espaço de expressões matemáticas para achar o modelo que melhor se encaixa

# 2 Desenvolvimento

Como dito anteriormente, regressão simbólica é um tipo de análise de regressão que utiliza um espaço de expressões matemáticas para achar o modelo que melhor se encaixa em uma dada função desconhecida. Em programação genética, existem diversas formar de representar um dado modelo. Mas a mais comum, e utilizada nesse trabalho, é a representação por árvores, fazendo a visualização e implementação serem mais visíveis.

## 2.1 Indivíduo

Cada indivíduo é representado por uma árvore. Nela, cada nó pode ser um operador matemático ou um terminal. Devido à essa representação, quanto maior a altura máxima da árvore, maior a complexidade das expressões.

Um exemplo de indivíduo é representado a seguir:

colocar exemplo de árvore

### 2.1.1 Operadores

Os operadores matemáticos possuem diferentes aridades, ou seja, possuem números de parâmetros diferentes. Cada um desses operadores interferem diretamente no resultado da expressão. Por exemplo, operadores como seno e cosseno possuem resultado limitado entre -1 e 1, enquanto operadores como exponencial alcançam valores extremamente altos.

### 2.1.2 Terminais

Os terminais são valores fixos na expressão, que podem ser dados por números reais ou por valores extraídos da função a ser encontrada. Enquanto os números

reais aleatórios não deveriam afetar muito o resultado da expressão, os valores extraídos são de suma importância para alcançar o valor adequado da função, já que foram extraídos da mesma.

## 2.2 Validação de um indivíduo

Para validar um indivíduo, deveria-se respeitar a aridade dos operadores e as folhas deveriam ser sempre operadores, já que operadores não podem possuem parâmetros vazios. Além disso, alguns dos operadores não aceitam certos valores por parâmetros, como é o caso do operador da divisão que não aceita divisão por 0 ou do operador logaritmo que não aceita valores negativos. Além disso, cada indivíduo deveria respeitar o tamanho máximo da árvore definido para que não houvessem indivíduos crescendo exponencialmente e atrapalhando a convergência dos mesmos.

## 2.3 Operadores genêticos

Os operadores genéticos são utilizados para alterar os indivíduos da população no intuito de melhorá-los a cada geração e encontrar o 'indivíduo ideal'.

### 2.3.1 Crossover

Comumente chamados de recombinação é um operador genético usado para variar a programação de um cromossomo de uma geração para a próxima. Uma recombinação é um processo de se pegar mais de uma solução progenitora e produzir uma solução descendente a partir deles.

Um exemplo de crossover entre duas árvores é representado a seguir:

colocar crossover árvore

Existem algumas formas de métodos de seleção de cromossomos para o cruzamento.

### 2.3.2 Mutação

A mutação leva em conta uma probabilidade de mutação para mutar cada gene de um indivíduo. Nesse caso, ao aplicarmos mutação, estaremos remodelando um galho da árvore representando uma função, inserindo uma subárvore de tamanho aleatório no lugar de um antigo galho do indivíduo.

Um exemplo de mutação é representado a seguir:

exemplo mutação

# 3 Implementação

A implementação foi feita em python. A escolha dessa linguagem foi devido à facilidade de manuseio de estruturas de dados como árvores.

## 3.1 Indivíduo

O indivíduo foi criado utilizando a biblioteca anytree de python. Essa biblioteca facilita a implementação de árvores bem como sua visualização.

Cada indivíduo armazena o nó de cada nível, além da aridade de cada nó, o número de filhos e um id único.

## 3.2 População inicial

Após ter sido definida a estrutura dos indivíduos, gera-se a população inicial. Foi utilizado o método 'grow' para criar os indivíduos. Esse método utiliza de uma probabilidade aleatória para gerar terminais ou operadores, fazendo com que as árvores tenham tamanhos diversos, aumentando a diversidade.

mostrar alguns indivíduos gerados

## 3.3 Função de fitness

Para avaliar um indivíduo, no caso de regressão simbólica, deve-se calcular a fitness do mesmo. Essa fitness pode ser calculada de diversas formas, mas a forma utilizada no trabalho prático atual foi a chamada 'NRMSE'.

mostrar fórmula NRMSE

## 3.4 Escolha de indivíduos

Após gerada a população inicial, e durante as gerações, deve-se selecionar dois indivíduos para próxima etapa. Foi utilizado o método de seleção por torneio, que consiste em selecionar o melhor indivíduo dentre 'k' indivíduos aleatórios com base na fitness. Essa seleção, se utilizado um 'k' pequeno, favorece a 'explotation' e pode ajudar em encontrar uma melhor solução para o problema.

## 3.5 Crossover

Realiza-se o crossover entre dois indivíduos. Extrai uma subárvore entre os três últimos níveis de cada gene, pois, de acordo com a literatura, pouco material genético deveria ser compartilhado para não afetar muito o comportamento da população. Dessa forma, também era evitado que os indivíduos passassem a aumentar de tamanho já que subárvores de mesma altura era transmitidas.

## 3.6 Mutação

Gera-se uma subárvore aleatória de tamanho definido e insere no lugar de algum nó da árvore. Também é evitado que indivíduos cresçam, substituindo a subárvore gerada por um nó de altura igual à altura da subárvore.

## 3.7 Nova população

Após terem sido aplicados os operadores genéticos citados anteriormente, deve-se adicionar o indivíduo à nova população. Essa adição foi feita de duas formas:

#### 3.7.1 Sem elitismo

Após terem sido aplicados os operadores, sempre adiciona o indivíduo resultante à nova população.

#### 3.7.2 Com elitismo

Só adiciona o indivíduo filho à nova população caso esse indivíduo possua fitness melhor que o pai.

# 4 Experimentos

Para avaliar o funcionamento do algoritmo foram realizados diversos testes com as bases de dados disponíveis. Cada um desses testes utilizou diferentes tipos de parametrização, no intuito de verificar como cada variável do algoritmo interferia no resultado final.

Como grande parte dos valores gerados pelo algoritmo são randômicos, como a criação das árvores, escolha de operadores, entre outros, foi utilizado uma seed para que todos os resultados randômicos tivessem o mesmo padrão, fazendo ser possível uma melhor análise dos resultados.

## 4.1 Elitismo

Primeiramente foi feito um teste para testar a influência do elitismo. Para esse teste, os parâmetros foram fixados da seguinte forma:

- Geração: 50
- Probabilidade de crossover: 0.9
- Probabilidade de mutação: 0.1
- Tamanho máximo dos indivíduos: 4
- Tamanho do torneio: 2

Os testes bases foram feitos de acordo com a Tabela 1.

| População | Geração | Probabilidade de crossover | Torneio | Elitismo |
|---|---|---|---|---|
| 150 | 50 | 0.9 | 2 | True |
| 150 | 50 | 0.9 | 2 | False |
| 300 | 50 | 0.9 | 2 | True |
| 300 | 50 | 0.9 | 2 | False |
| 450 | 50 | 0.9 | 2 | True |
| 450 | 50 | 0.9 | 2 | False |
| 600 | 50 | 0.9 | 2 | True |
| 600 | 50 | 0.9 | 2 | False |
| 750 | 50 | 0.9 | 2 | True |
| 750 | 50 | 0.9 | 2 | False |

Table 1: My table

Esses testes foram feitos no dataset synth1 e resultaram nos seguintes valores de fitness média por execução:

| População | Elitismo | Melhor Fitness | Pior Fitness | Fitness Média |
|---|---|---|---|---|
| 150 | False | 0.344403723 | 39.736031738 | 0.76229664 |
| 300 | False | 0.444181580 | 21.715649614 | 0.626452953 |
| 450 | False | 0.335857646 | 6.1422037383 | 0.515225522 |
| 600 | False | 0.1109138513 | 16.18509664 | 0.479958947 |
| 750 | False | 0.2110320063 | 108.4944595 | 0.308710017 |

Table 2: Execução do algoritmo sem elitismo

| População | Elitismo | Melhor Fitness | Pior Fitness | Fitness Média |
|---|---|---|---|---|
| 150 | True | 0.624727633588311 | 1.1641793421718536 | 0.890964018237252 |
| 300 | True | 0.26461710389283016 | 0.2735141448942379 | 0.2734844880908999 |
| 450 | True | 0.06655913723097262 | 0.06837350416027892 | 0.06687362749871904 |
| 600 | True | 0.04419758026640803 | 0.05181260653376336 | 0.051760894596640745 |
| 750 | True | 0.30082681842133063 | 0.3012763669483054 | 0.30107389930454087 |

Table 3: Execução do algoritmo com elitismo

Após esses testes, verificou-se que, utilizando elitismo, os resultados se apresentaram expressivamente melhores do que a não utilização dessa técnica. Além disso, foi observado que, não utilizando o elitismo, o indivíduo de pior fitness apresentava uma fitness muito pior do que a média dos indivíduos daquela população. Isso pode ser explicado pelo fato da mutação gerar subárvores aleatórias que prejudicam o resultado da expressão.

Outro ponto importante é que, ao utilizar elitismo, a média dos indivíduos se mantém muito próxima do pior e do melhor indivíduos da população. Isso evindencia o fato do elitismo fazer pouca exploração, não favorecendo indivíduos piores do que os atuais mas que potencialmente levariam à uma solução melhor

no futuro, como é o caso do último teste da tabela 3 em que a fitness dos indivíduos se apresentaram ruins comparados com os testes realizados em outras populações. Nesse caso, o algoritmo fez pouca exploração e alcançou somente um ótimo local, distante do ótimo global.

## 4.2 Tamanho máximo dos indivíduos

O próximo teste realizado teve o intuito de verificar a interferência do tamanho dos indivíduos para o resultado final. Os testes foram feitos na base de dados synth1 e synth2, pois, como as bases de dados possuem funções de complexidade diferentes, seria adequado verificar como o tamanho influenciaria de acordo com a complexidade. Os parâmetros do teste estão listados a seguir:

- Geração: 50
- Probabilidade de crossover: 0.9
- Probabilidade de mutação: 0.1
- Elitismo: Sim
- Tamanho do torneio: 2
- Quantidade de indivíduos por população: 350

| Tamanho dos indivíduos | Melhor Fitness | Pior Fitness | Fitness Média |
|---|---|---|---|
| 3 | 0.12359172703932719 | 0.12359172703932719 | 0.12359172703932719 |
| 4 | 0.12359172703932719 | 0.12359172703932719 | 0.12359172703932719 |
| 5 | 12358906586614692 | 12358906586614692 | 12358906586614692 |
| 6 | 0.12359172703932718 | 0.12359172703932718 | 0.12359172703932718 |
| 7 | 0.12359129523367042 | 0.12359129523367042 | 0.12359129523367042 |
| 8 | 0.1235912952336704 | 0.1235912952336704 | 0.1235912952336704 |
| 9 | 0.12359172703932718 | 0.12359172703932718 | 0.12359172703932718 |

| Tamanho dos indivíduos | Melhor Fitness | Pior Fitness | Fitness Média |
|---|---|---|---|
| 3 | 0.7066542311785389 | 0.7667272500226958 | 0.7619653183264381 |
| 4 | 0.6111207832180868 | 0.6335470925984652 | 0.6113351102298297 |
| 5 | 0.5774455207208067 | 0.5980808316823124 | 0.5856659225014924 |
| 6 | 0.6254560541773741 | 0.6555473358684234 | 0.6546684052167549 |
| 7 | 0.55244286781125015 | 0.55244286781125015 | 0.55244286781125004 |
| 8 | 0.5903533341555821 | 0.6017689096021673 | 0.5970521615834704 |
| 9 | 0.49624029073058645 | 0.49624029073058645 | 0.49624029073058634 |

Com os resultados apresentados verificou-se que, usando a base de dados synth1, a fitness permaneceu quase que inalterada com diferentes tamanhos de árvore. Isso pode ser explicado, pois como a função a ser encontrada é de certa

forma simples, uma árvore de tamanho pequeno já se mostra suficiente para encontrar uma solução razoável para o problema. Enquanto isso, para a base de dados synth2 verificou-se que, a fitness dos indivíduos diminuía inversamente proporcional ao aumento da árvore, demonstrando que, por ser uma função mais complexa, árvores maiores conseguem alcançar uma aproximação melhor do que árvores pequenas.

## 4.3 Quantidade de indivíduos da população

Esse teste foi realizado para validar a hipótese de que, quanto mais indivíduos na população, maior a probabilidade de obter indivíduos que alcancem a função objetiva. Dessa forma, era esperado que, quanto maior a população, menor seria a fitness encontrada. Esse teste foi feito utilizando a base de dados synth2, já que é uma base de dados com função objetiva mais complexa que a outra base de dados sintética e demoraria mais a convergir mesmo com uma população inicial alta, ao contrário da synth1. Os parâmetros fixados para esses teste foram:

- Geração: 50
- Probabilidade de crossover: 0.9
- Probabilidade de mutação: 0.1
- Tamanho dos indivíduos: 5
- Elitismo: Sim
- Tamanho do torneio: 2

| Quantidade de indivíduos | Melhor Fitness | Pior Fitness | Fitness Média |
|---|---|---|---|
| 150 | 0.8330162555941489 | 0.8330162555941489 | 0.8330162555941489 |
| 250 | 0.6176124426359278 | 0.6439223941937607 | 0.6417924519422588 |
| 350 | 0.5404834934950826 | 0.5586475273861882 | 0.5581181038145426 |
| 400 | 0.5926740564339222 | 0.610140721279847 | 0.6074369578446666 |
| 450 | 0.4665610046808454 | 0.4736387716776842 | 0.4700971022630252 |
| 500 | 0.5011806454501798 | 0.5170298182752043 | 0.5129270685584074 |
| 550 | 0.4779723694899959 | 0.4854228233516763 | 0.4823488738124393 |

Como pode ser observado pela tabela, o aumento da quantidade de indivíduos por população, apesar que em algumas ocasiões gerou resultados piores do que populações anteriores, gerou uma grande melhora para o desempenho do algoritmo, comprovando a hipótese inicial.

## 4.4 Tamanho do Torneio

O próximo teste teve como objetivo avaliar a influência do tamanho do torneio no resultado. Esse teste foi feito utlizando a base de dados synth1. Os parâmetros desse teste foram:

- Geração: 50
- Tamanho da população: 350
- Probabilidade de crossover: 0.9
- Probabilidade de mutação: 0.1
- Tamanho dos indivíduos: 4
- Elitismo: Sim

| Tamanho do torneio | Melhor Fitness | Pior Fitness | Fitness Média |
|---|---|---|---|
| 1 | 0.3315619876700638 | 0.3371267418569974 | 0.3361267418569974 |
| 3 | 0.12359178314755569 | 0.12359178314755569 | 0.12359178314755569 |
| 5 | 0.2558410690994585 | 0.2558410690994585 | 0.2558410690994584 |
| 7 | 0.05454430660123331 | 0.06552971330149143 | 0.06480373134776272 |
| 9 | 0.06655114861993645 | 0.06655114861993645 | 0.06655114861993647 |
| 11 | 0.06655114910282602 | 0.06655114910282602 | 0.06655114910282602 |
| 13 | 0.06654903939933345 | 0.06654903939933345 | 0.06654903939933345 |
| 15 | 0.04253103566819662 | 0.04253103566819662 | 0.04253103566819661 |

Como pode ser observado, à medida que aumenta o tamanho do torneio, a fitness dos indivíduos diminuem, exceto algumas exceções. Isso pode ser explicado, pois, ao escolher um indivíduo aleatório mais vezes, a probabilidade de selecionar um indivíduo com fitness boa aumentam, reproduzindo tais indivíduos. De igual maneira, aumentar o tamanho do torneio pode ocasionar a seleção de uma melhor fitness localmente, dificultando encontrar a melhor fitness global. Mas isso não foi verificado pelos testes.