

# Algoritmos Evolutivos para Regressão Simbólica utilizando a estrutura Interação-Transformação

*Guilherme Seidyo Imai Aldeia*  
*Prof. Dr. Fabrício Olivetti de França*

Universidade Federal do ABC  
Centro de Matemática, Computação e Cognição  
Bacharelado em Ciência da Computação

Santo André  
2019

# Índice

## ① Introdução

Análise de regressão

Regressão simbólica e algoritmos genéticos

# Tipos de regressão

Técnicas de regressão buscam encontrar relações entre variáveis explanatórias  $\mathbf{x}$ ,  $\mathbf{x} = (x_1, x_2, \dots, x_{n-1}, x_n)$  e uma variável alvo  $y$ . São classificadas como:

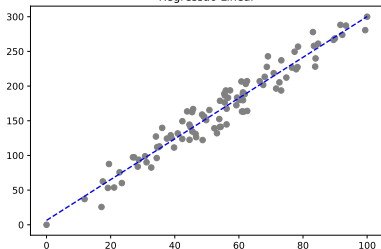
- **Paramétricas** - partem de funções previamente estabelecidas e ajustam parâmetros livres dessa função;
- **Não-paramétricas** - ajustam tanto os parâmetros livres quanto a própria função.

# Regressões paramétricas

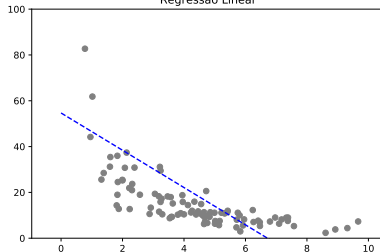
- ✓ Apresentam uma maior simplicidade.
- ✓ Bons resultados quando os dados se comportam de acordo com a função utilizada.
- ✗ Limitação quanto às formas de função, menor expressividade.

# Regressão linear e Regressão polinomial

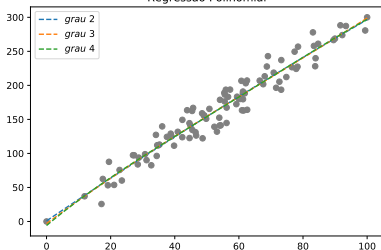
Regressão Linear



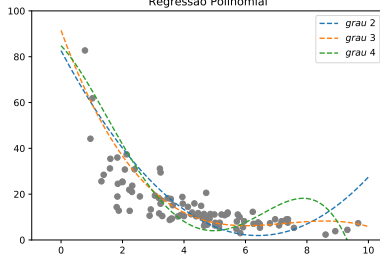
Regressão Linear



Regressão Polinomial



Regressão Polinomial

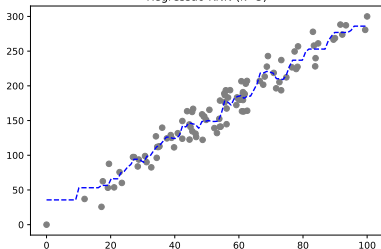


# Regressões não-paramétricas

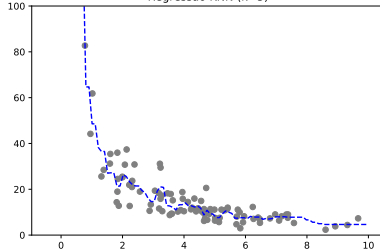
- ✓ Abordagem mais flexível.
- ✓ Resultados mais expressivos, sem estarem atrelados à uma função prévia.
- ✗ Maior complexidade para obter um bom resultado.

# Regressão KNN e Regressão Simbólica

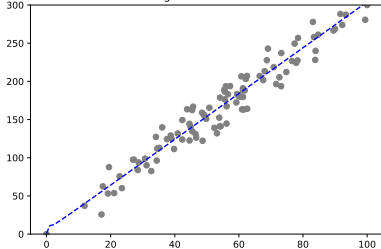
Regressão KNN (k=5)



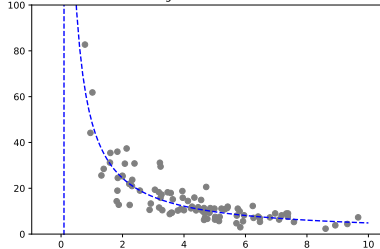
Regressão KNN (k=5)



Regressão Simbólica



Regressão Simbólica



# Índice

## ① Introdução

Análise de regressão

Regressão simbólica e algoritmos genéticos



# Regressão simbólica

Costuma ser implementada através da **programação genética**, evoluindo populações de soluções.

- Estratégia de otimização bioinspirada;
- Busca aleatória direcionada, guiada por uma função de afinidade (*fitness*);
- Pode obter soluções complexas em um amplo espaço de busca.

# Algoritmos genéticos

Inspirados na teoria da evolução de Darwin:

- **Variação** - Existem pequenas variações dentro da população, algumas resultam em uma maior aptidão ao meio;
- **Seleção** - os recursos são limitados, havendo uma competição dentro da própria população, favorecendo os mais aptos - seleção natural;
- **Herança** - Os indivíduos que forem capazes de se reproduzir terão seus traços herdados pelas futuras gerações.

# Estrutura geral

---

**Algoritmo 1:** Estrutura geral de um algoritmo genético.

---

**Entrada:** Parâmetros de execução, critério de parada

**Saída:** função simbólica  $f$

$P \leftarrow [n \text{ Soluções aleatórias}];$

**para**  $g$  *Gerações* **faça**

$P' \leftarrow \text{Crossover}(P);$

$P' \leftarrow \text{Mutação}(P');$

$P \leftarrow \text{Seleção}(P');$

**se** *critério de parada foi atingido* **então**

        Parar evolução;

**fim**

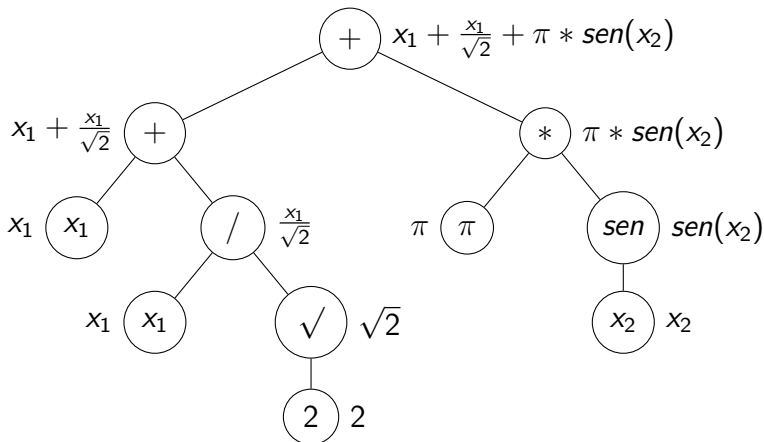
**fim**

**retorna**  $\text{maxFitness}(P);$

---

# Representação em árvores

No caso da regressão simbólica, as soluções são funções matemáticas que são representadas por **árvores**.



# aspectos positivos

- ✓ Grande poder de exploração, evitando atração para o ótimo local mais próximo da região inicial.
- ✓ Simplicidade de implementação.
- ✓ Bons resultados na literatura - diversos exemplos bem sucedidos de aplicação nos mais diversos problemas.

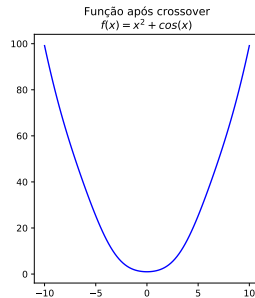
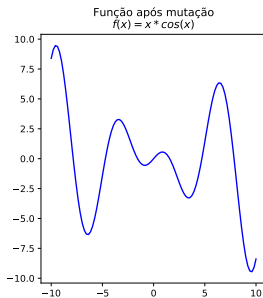
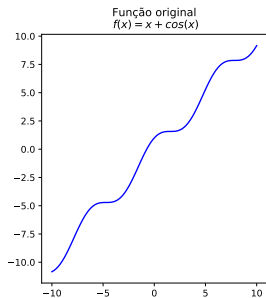
# Aspectos negativos

A regressão simbólica por meios da programação genética apresenta diversos problemas, associados à forma em que as soluções são representadas, através de árvores.

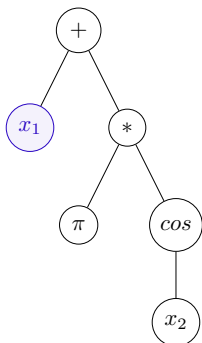
- ✗ Falta de transição suave.
- ✗ Espaço de busca infinito, podendo conter redundâncias ou expressões de baixa interpretabilidade.

.

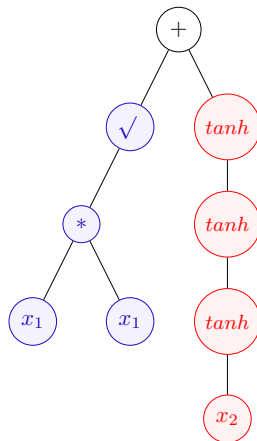
# Aspectos negativos



# Aspectos negativos



$$f(x_1, x_2) = x_1 + \pi * \cos(x_2)$$



$$f(x_1, x_2) = \sqrt{x_1 * x_1} + \tanh(\tanh(\tanh(x_2)))$$



# Alternativas

Para minimizar os problemas, podemos:

- Restringir o espaço de busca;
- Priorizar funções simples;
- Mudar a construção e representação de expressões.

# Motivação

- Atender aos pontos para minimizar os problemas da programação genética pode levar à algoritmos com um melhor desempenho nas tarefas de predizer e inferir.
- Também é desejado soluções que minimizem o erro na predição e apresentem resultados competitivos.

*O estado-da-arte em regressão simbólica apresenta resultados competitivos porém ainda possui limitações frente a alternativas.*

# Índice

## ③ Objetivos

### Objetivos

# Objetivos gerais

Propor um algoritmo evolutivo de regressão simbólica utilizando uma representação alternativa, comparando seu desempenho com algoritmos bem sucedidos de *machine learning*.

# Objetivos específicos

- Criação de uma estrutura de dados que representa cada indivíduo da população;
- Definição de uma função de afinidade para guiar a busca;
- Criação de um algoritmo de mutação;
- Criação de um algoritmo de cruzamento.

# Índice

## ④ Proposta

Estrutura Interação-Transformação

Operadores evolutivos

Algoritmos implementados

# Utilizando outra representação

Em "*A greedy search tree heuristic for symbolic regression*" é proposta uma nova representação que atende esses requisitos — a representação **Interação-Transformação** (IT).

Esta representação naturalmente introduz uma restrição no espaço de busca, priorizando funções simples.

# Estrutura Interação-Transformação

A estrutura IT descreve a aplicação de uma função de transformação sobre funções polinomiais das variáveis originais.



# Estrutura Interação-Transformação

Sejam:

- $n$  o número de atributos do problema de regressão;
- $(\mathbf{x}, y)$ , com  $\mathbf{x} = (x_1, x_2, \dots, x_n)$  um ponto no domínio do problema;
- $\mathbf{k}$  um vetor de expoentes  $\mathbf{k} \in \mathbb{Z}^n$ ;
- $t$  uma função unária  $t : \mathbb{R} \rightarrow \mathbb{R}$  (chamada de função de transformação);
- $i(\mathbf{x}, \mathbf{k})$  uma função definida como o produto de cada variável elevada ao expoente de  $\mathbf{k}$  de mesmo índice (chamada de função de interação):

$$i(\mathbf{x}, \mathbf{k}) = \prod_{i=1}^n x_i^{k_i}. \quad (1)$$

# Estrutura Interação-Transformação

Uma estrutura Interação-Transformação (IT) é uma tupla  $(t, \mathbf{k})$  onde  $t$  é qualquer função matemática unária e  $\mathbf{k}$  um vetor de dimensão  $n$  contendo expoentes aplicados às respectivas variáveis  $x_i$ .

$$it(\mathbf{x}) = t(i(\mathbf{x}, \mathbf{k})). \quad (2)$$

# Estrutura Interação-Transformação

Uma única tupla é chamada de **termo IT**, e uma combinação linear de  $m$  termos é chamada de *expressão IT*.

$$Expr\ it(\mathbf{x}) = \sum_{i=1}^m w_i \cdot it_i(\mathbf{x}), \quad (3)$$

onde  $\mathbf{w}$  representa um vetor de coeficientes (pesos) dos termos IT de mesmo índice. Os valores de  $\mathbf{w}$  podem ser ajustados com algum método de regressão linear.

# Exemplo

Para ilustrar, vamos ver como compor a expressão

$$f(x_1, x_2) = x_1 + \pi * \cos(x_2):$$

$$\begin{aligned} it_1 &= (id, [1,0]), \\ it_2 &= (\cos, [0,1]), \\ \mathbf{w} &= [1, \pi]. \end{aligned} \tag{4}$$

Que resulta em:

$$f(x_1, x_2) = 1 * id(x_1^1 \cdot x_2^0) + \pi * \cos(x_1^0 * x_2^1). \tag{5}$$

# Exemplos representáveis e não representáveis

---

---

Expressões representáveis

$$x_1 + \pi * \cos(x_2)$$

$$x_1 + x_2 + x_1 * x_2$$

$$10 * \log\left(\frac{l}{l_0}\right) + 5.0$$

$$a^2 + 2 * a * b + b^2$$

Expressões não representáveis

$$\sqrt{x_1 * x_1} + \tanh(\tanh(\tanh(x_2)))$$

$$\frac{1}{\frac{1}{R_1} + \frac{1}{R_2}}$$

$$\frac{\sin(x_1 * x_2)}{x_3}$$

$$\sqrt{a^2 + b^2}$$

---

---

# Índice

## ④ Proposta

Estrutura Interação-Transformação

**Operadores evolutivos**

Algoritmos implementados

# Operadores evolutivos

- A implementação requer métodos que manipulem a representação computacional da estrutura.
- **Operadores evolutivos** — operadores modificam as estruturas (que dizem respeito à forma da função, não seus parâmetros livres) de **expressões IT**.

Vamos apresentar os operadores com potencial de melhora na população.

# Operador de inicialização

- Preenche uma população inicialmente vazia com soluções aleatórias.
- Faz o tratamento para evitar duplicata de termos — essa redundância aumenta a expressão sem agregar novas características.
- Também cuida para não existirem termos nulos ( $\mathbf{k} = \mathbf{0}^n$ ) ou expressões vazias.



# Operador de inicialização

---

**Algoritmo 1:** Operador de origem (método *oporigem*).

---

**Entrada:** *pop\_len*: Tamanho da população  
           *n\_vars*: Número de variáveis explicatórias  
           *funcs*: Conjunto de possíveis funções de transformação  
**Saída** : *P*: População com soluções aleatórias

$P \leftarrow [ ]$ ;

**repita**

$n\_terms \leftarrow \text{random}([1, \dots, 4])$ ;  
      $terms \leftarrow [[\text{random}([0, \dots, 4]) \text{ para } _ \in n\_vars] \text{ para } _ \in n\_terms]$ ;  
      $funcs \leftarrow [\text{random}(funcs) \text{ para } _ \in n\_terms]$ ;  
      $P \leftarrow P + [(terms, funcs)]$ ;

**até**  $\text{len}(P)$  seja igual a *pop\_len*;

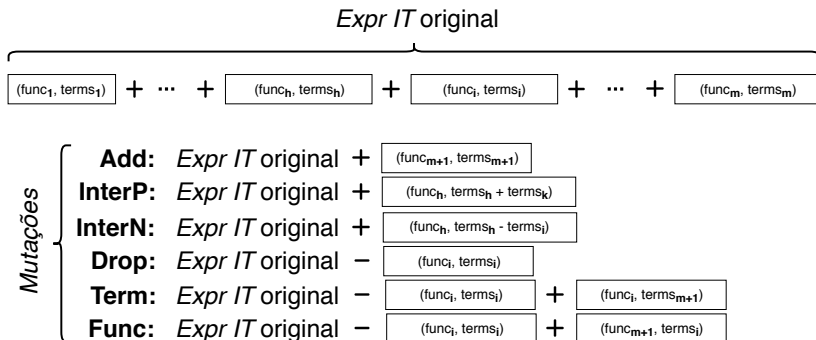
**retorna** *P*;

---

# Operador de mutação

- Introduz variação e ajuda a contornar os mínimos locais.
- Modifica apenas os aspectos da função (*terms*, *funcs*), sem modificar os parâmetros livres da expressão (*coeffs*, *intercept*).
- 4 grupos de mutação:
  - **Expansores de expressão:** *Add*, *Interp*, *Intern*;
  - **Redutores de expressão:** *Drop*;
  - **Modificadores de expoentes:** *Term*;
  - **Modificadores de termos:** *Func*.
- Tamanho mínimo e máximo das soluções é controlado.

# Operador de mutação



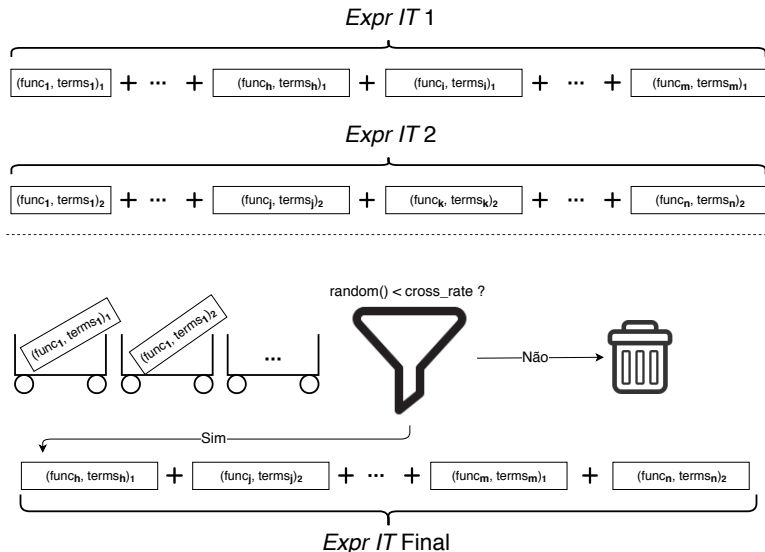
# Operador de seleção

- Impõe uma pressão seletiva na população, dando mais chances para as soluções de melhor aptidão.
- Utiliza uma função de torneio para comparar expressões e selecionar a melhor.

# Operador de *crossover*

- Simula a troca de informação genética da reprodução sexuada.
- Seleciona duas funções e percorre cada uma, sorteando para cada termo IT se ele irá compor a expressão filha.

# Operador de *crossover*



# Índice

## ④ Proposta

Estrutura Interação-Transformação

Operadores evolutivos

Algoritmos implementados

# Algoritmos implementados

- **IT-MUT**: Algoritmo que utiliza apenas a mutação;
- **IT-CX**: Algoritmo evolutivo que utiliza apenas o *crossover*;
- **ITEA**: Algoritmo que utiliza a combinação da mutação e do *crossover*.



# Índice

## ⑤ Métodos Experimentais

Avaliação

Ajuste dos modelos

Benchmark

Comparação com outras técnicas

# Como avaliar os algoritmos desenvolvidos

- Uso de métricas populares para medir o desempenho de algoritmos de regressão e *machine learning*, utilizando várias bases de dados diferentes;
- Análise em termos de simplicidade dos resultados;
- Comparação com algoritmos populares de aprendizagem de máquina.

# Métricas para comparação

- **RMSE:** eleva os erros ao quadrado antes de utilizá-los no cálculo da média — se o *erro*  $> 1$  ele é amplificado, caso contrário é reduzido.
- **Mediana:** Mais robusta à *outliers*, inevitáveis na computação evolutiva.

# Índice

## ⑤ Métodos Experimentais

Avaliação

Ajuste dos modelos

Benchmark

Comparação com outras técnicas

# Hiper-parâmetros

Os algoritmos possuem vários *hiper-parâmetros* — que não são ajustados pelo modelo e que influenciam na busca e no resultado.

- **Mutações:** será testado cada grupo isoladamente e, após isso, será feita a combinação dos dois melhores grupos até que todos tenham sido combinados.
- **Crossover:** será feito o teste para os valores de 0.1 até 0.7, com um passo de 0.1.

Obtidos o melhor grupo de mutação e a melhor taxa de *crossover*, estes serão os parâmetros utilizados nos modelos para análise de seus desempenhos.

# Hiper-parâmetros

Algoritmo	Hiper-parâmetros
IT-MUT	Tamanho da população, Número de gerações, Conjunto de funções, <b>Mutações</b>
IT-CX	Tamanho da população, Número de gerações, Conjunto de funções, <b>Taxa de crossover</b>
ITEA	Tamanho da população, Número de gerações, Conjunto de funções, <b>Mutações,</b> <b>Taxa de crossover</b>

# Índice

## ⑤ Métodos Experimentais

Avaliação

Ajuste dos modelos

**Benchmark**

Comparação com outras técnicas

# Dados e estratégias

- Dados do mundo real;
- Diferentes tamanhos, número de atributos, relações entre as variáveis;
- Uso de divisão em *folds* (80%/20%) e validação cruzada;
- Repetição de 100 execuções ao todo/ 20 por *fold*.



# Bases de dados

Nome da base	# Partições	# Atributos	# Amostras (treino/validação)
Airfoil	5	6	1202/301
Concrete	5	9	824/206
Energy Cooling	5	9	614/154
Energy Heating	5	9	614/154
Tower Data	5	26	3999/1000
Wine Red	5	12	1279/319
Wine White	5	12	3919/979
Yatch	5	7	246/62

# Índice

## ⑤ Métodos Experimentais

Avaliação

Ajuste dos modelos

Benchmark

Comparação com outras técnicas

# Comparação com outras técnicas

- Comparação com algoritmos de regressão populares: **OLS**, **Ridge**, **LARS**, **Lasso**, **KNN**, **Bayesian**, **XGBoost**, **MLP**.
- Ajuste dos hiper-parâmetros de cada um utilizando o *grid-search*.
- Realizar a execução de todos no mesmo ambiente - desenvolvidos na linguagem *Python*.
- Aplicar as mesmas bases, mesmas divisões de *folds*, reportando as mesmas métricas, com o mesmo número de execuções.

# Pré processamento

- Feito apenas para os algoritmos comparados.
- Normalização *z-score* para todos - média 0 e desvio padrão 1, tirando a necessidade de ajuste de intercepto.
- Normalização *min-max* para a rede neural - devido às funções de ativação.

# Índice

## ⑥ Experimentos e resultados

### Testes preliminares

Melhor grupo de mutação

Melhor taxa de crossover

Variação na população e no número de indivíduos

Análise dos algoritmos desenvolvidos

Comparação com outras técnicas

# Definição dos hiper-parâmetros em comum

Teste inicial com 1000 gerações e 500 indivíduos para observar o desenvolvimento das expressões sem restrições:

- Adicionado um limite de expoentes:  $[-10,10]$ ; e limite tamanho da expressão:  $[3, 10]$ .
- Tamanho da população e número de gerações fixado em 100.
- Conjunto de funções utilizado contém apenas funções comumente encontradas.

Diminuição na complexidade das equações sem perda de generalidade, não impactando de forma significativa nos resultados finais mas fornecendo resultados mais interpretáveis.

# Definição dos hiper-parâmetros em comum

Dessa forma, temos para os hiper-parâmetros comuns aos três algoritmos, os valores:

- **Tamanho da população:** 100;
- **Número de gerações:** 100;
- **Conjunto de funções:**  
 $\{id, sen, cos, tanh, \sqrt{\phantom{x}}, log, log_{10}, exp\}.$

# Índice

## ⑥ Experimentos e resultados

Testes preliminares

Melhor grupo de mutação

Melhor taxa de crossover

Variação na população e no número de indivíduos

Análise dos algoritmos desenvolvidos

Comparação com outras técnicas



# Classificação dos grupos de mutação

Mediana do RMSE para a base de validação em 100 experimentos.

Base de dados	Add e Drop	Term	Intern e Interp	Func
Airfoil	3.269	4.065	3.488	4.66
Concrete	7.018	10.402	11.148	12.988
Energy Cooling	2.178	3.104	2.88	4.355
Energy Heating	1.501	2.702	2.921	4.512
Tower Data	35.689	35.926	39.762	61.12
Wine Red	0.641	0.656	0.681	0.739
Wine White	0.741	0.766	0.778	0.84
Yacht	1.854	2.182	1.248	5.505
Pontuação geral	1.125	2.375	2.5	4.0
Classificação final	1	2	3	4

# Classificação dos grupos de mutação

Mediana do RMSE para a base de validação em 100 experimentos.

Base de dados	Add, Drop e Term	Add, Drop, Term, Intern e Interp	Todos os grupos
Airfoil	3.183	2.893	2.932
Concrete	6.95	6.908	6.962
Energy Cooling	1.99	1.888	1.851
Energy Heating	1.274	1.035	1.107
Tower Data	34.374	31.32	32.196
Wine Red	0.637	0.635	0.64
Wine White	0.736	0.737	0.735
Yacht	1.221	0.849	0.871
Pontuação geral	2.625	1.375	2.0
Classificação final	3	1	2

# Melhor grupo de mutação

- Melhor grupo individual é *Add*, *Drop*; seguido do grupo *Term* - esses dois grupos foram combinados.
- Após isso, os dois melhores grupos foram *Add*, *Drop*, *Term* e *Intern*, *Interp*;
- O grupo *Func* foi a pior de todos os grupos individuais - atingindo um platô na geração 20 na maioria dos casos;
- Quando combinada com as outras, a *Func* piorou o desempenho do algoritmo;
- O grupo *Term* também se estabiliza em um ponto, mas com um resultado melhor que o grupo *Func*.

# Índice

## ⑥ Experimentos e resultados

Testes preliminares

Melhor grupo de mutação

Melhor taxa de crossover

Variação na população e no número de indivíduos

Análise dos algoritmos desenvolvidos

Comparação com outras técnicas

# Classificação das taxas de *crossover*

Mediana do RMSE para a base de validação em 100 experimentos.

Base de dados	Taxa 0.10	Taxa 0.20	Taxa 0.30	Taxa 0.40	Taxa 0.50	Taxa 0.60	Taxa 0.70
Airfoil	4.425	4.402	4.495	4.588	4.78	4.969	4.987
Concrete	11.817	11.503	11.889	12.36	13.05	13.46	13.772
Energy Cooling	3.781	3.747	3.708	3.945	4.082	4.648	4.779
Energy Heating	3.723	3.692	3.733	3.948	4.295	4.677	5.372
Tower Data	58.865	58.821	59.317	59.337	60.073	60.946	61.543
Wine Red	0.737	0.731	0.733	0.74	0.751	0.749	0.762
Wine White	0.834	0.831	0.835	0.841	0.845	0.85	0.851
Yacht	4.0	4.017	4.264	4.451	5.388	5.815	5.968
Pontuação geral	2.125	1.25	2.625	4.0	5.125	5.875	7.0
Classificação final	2	1	3	4	5	6	7

# Melhor taxa de *crossover*

- A taxa que obteve o melhor resultado foi 0.2 (20%);
- Quanto maior a taxa menor a classificação final;
- A taxa de *crossover* tem um reflexo direto no tamanho da expressão filha;
  - Para taxas pequenas, é esperado que poucos ou quase nenhum termo IT seja herdado.
- Convergência inferior ao IT-MUT.

# Índice

## ⑥ Experimentos e resultados

Testes preliminares

Melhor grupo de mutação

Melhor taxa de crossover

Variação na população e no número de indivíduos

Análise dos algoritmos desenvolvidos

Comparação com outras técnicas

# Variando o tamanho da população

- Determinados os melhores hiper-parâmetros para os algoritmos, uma pequena análise da variação do tamanho da população e do número de gerações foi feita.
- Os valores anteriores foram no mínimo dobrados; mas sem aumentar muito o tempo necessário para execução.



# Variando o tamanho da população

- O aumento do tamanho da população desloca a convergência um pouco para baixo em **todos** os casos em relação à configuração original de 100 indivíduos e 100 gerações;
- O aumento do número de gerações sobrepõe a linha contínua e complementa-a por mais 200 gerações.

# Melhores resultados

- Apenas a variação no grupo de mutação e taxa de *crossover* foi feita - o ITEA utiliza a combinação das melhores configurações individuais.
- Os melhores parâmetros foram definidos como:
  - IT-MUT apresenta um melhor desempenho com 250 gerações (e 100 indivíduos);
  - e os algoritmos ITEA e IT-CX com 300 indivíduos (e 100 gerações).

# Índice

## ⑥ Experimentos e resultados

Testes preliminares

Melhor grupo de mutação

Melhor taxa de crossover

Variação na população e no número de indivíduos

Análise dos algoritmos desenvolvidos

Comparação com outras técnicas

# Tamanho das expressões

Base de dados	IT-MUT	IT-CX	ITEA
Airfoil	$144.64 \pm 13.624$	$30.9 \pm 11.228$	$64.26 \pm 14.854$
Concrete	$181.01 \pm 17.269$	$41.9 \pm 16.572$	$71.48 \pm 17.932$
Energy Cooling	$171.37 \pm 20.722$	$43.77 \pm 12.321$	$64.46 \pm 18.696$
Energy Heating	$173.93 \pm 18.829$	$47.53 \pm 12.771$	$61.14 \pm 15.062$
Tower Data	$738.1 \pm 72.669$	$112.78 \pm 45.133$	$188.73 \pm 98.793$
Wine Red	$291.88 \pm 33.127$	$64.44 \pm 23.47$	$96.03 \pm 36.084$
Wine White	$288.86 \pm 33.362$	$70.53 \pm 20.107$	$92.58 \pm 29.14$
Yacht	$166.97 \pm 21.292$	$23.38 \pm 13.485$	$53.43 \pm 14.911$
Pontuação geral	3.0	1.0	2.0
Classificação final	3	1	2

# Tamanho das expressões - base Airfoil

Expressão de menor RMSE (2.250) para o IT-MUT (tamanho=144)

```
1.471525*log1p(x3^-3 * x4^-2) +  
0.607964*log1p(x3^7 * x4^-4) +  
-0.954491*log1p(x0^-9 * x1^2 * x2^-7 * x3^3 * x4^-6) +  
-0.513344*log1p(x0^8 * x2^7 * x3^-4 * x4^3) +  
-1.42186*log1p(x0^-8 * x2^2 * x3^2 * x4^-9) +  
0.445444*log1p(x0^-8 * x1^3 * x2^-7 * x3^6 * x4^-3) +  
-3.310411*log1p(x0^-2 * x2^-3 * x3^2 * x4) +  
-2740.416799*log1p(x1 * x4^3) +  
0.011361*id(x0 * x4) +  
-0.141906*sqrt(x0) +  
110.57095766780208
```

# Tamanho das expressões - base Airfoil

Expressão de menor RMSE (3.905) para o IT-CX (tamanho=27)

$$\begin{aligned} & -2.440403 \cdot \log_{10}(x_0^3 * x_2^3 * x_4^3) + \\ & 9.811232 \cdot \log_{10}(x_1^2 * x_2^3) + \\ & 128.62804941792177 \end{aligned}$$

# Tamanho das expressões - base Airfoil

Expressão de menor RMSE (3.179) para o ITEA (tamanho=86)

```
0.979865*log1p(x0^-6 * x2^-8 * x3^4 * x4^-8) +  
-0.512595*log1p(x0^8 * x2^-4 * x3^-2 * x4^-4) +  
-2.138797*log1p(x0^-8 * x2^-4 * x3^6 * x4^-4) +  
-151537286637.53867*log1p(x2^2 * x3^-4 * x4^2) +  
1.438948*log1p(x0^2 * x2 * x3 * x4) +  
123.3527228794443
```

# Tamanho das expressões

- É possível realizar uma inferência com muito mais facilidade para a expressão encontrada pelo IT-CX, que apresenta maior simplicidade, porém esse é o resultado de maior erro associado.
- O resultado encontrado pelo IT-MUT apresenta o menor erro entre eles, mas uma função mais complexa.
- O ITEA entrega um resultado intermediário entre os dois, tanto no erro da predição quanto no tamanho — evidência de que há um *payoff* entre minimização do erro e minimização da complexidade das expressões.



# RMSE Médio

Base de dados	ITEA-MUT	ITEA-CX	ITEA
Airfoil	$2.758 \pm 0.433$	$4.335 \pm 0.163$	$3.66 \pm 0.209$
Concrete	$6.624 \pm 0.31$	$11.099 \pm 0.843$	$8.116 \pm 0.434$
Energy Cooling	$1.692 \pm 0.158$	$3.555 \pm 0.199$	$3.023 \pm 0.216$
Energy Heating	$0.687 \pm 0.129$	$3.447 \pm 0.32$	$2.605 \pm 0.198$
Tower Data	$29.295 \pm 2.043$	$58.27 \pm 2.534$	$41.62 \pm 4.541$
Wine Red	$0.676 \pm 0.255$	$0.722 \pm 0.039$	$5402.607 \pm 53748.761$
Wine White	$0.745 \pm 0.052$	$0.821 \pm 0.019$	$0.75 \pm 0.013$
Yacht	$0.904 \pm 0.691$	$3.212 \pm 1.158$	$1.166 \pm 0.156$
Pontuação geral	1.0	2.875	2.125
Classificação final	1	3	2

# Desempenho final

O IT-MUT é bom quando o foco está na minimização de erro, seguido do algoritmo ITEA, e por fim do algoritmo IT-CX. Esse resultado é particularmente interessante devido à simplicidade de implementação de um algoritmo baseado apenas em uma mutação.

# Índice

## ⑥ Experimentos e resultados

Testes preliminares

Melhor grupo de mutação

Melhor taxa de crossover

Variação na população e no número de indivíduos

Análise dos algoritmos desenvolvidos

Comparação com outras técnicas

# Comparação com outras técnicas

- Essa comparação será feita em termos de minimização de erro, pois é um dos objetivos mais almejados, utilizando a métrica RMSE.
- Apenas o melhor dos algoritmos desenvolvidos (em termos de minimização de erro) será comparado - o IT-MUT, em sua melhor configuração:

Hiper-parâmetro	Melhor configuração
Tamanho da população	100
Número de gerações	250
Conjunto de funções	$\{id, sen, cos, tanh, \sqrt{\cdot}, log, log1p, exp\}$
Mutações	Add, Drop, Intern, Interp, Term

# Desempenho dos algoritmos comparados

Base de dados	IT-MUT	Bayesian	KNN	LARS lasso
Airfoil	$2.758 \pm 0.433$	$4.819 \pm 0.156$	$2.628 \pm 0.095$	$4.819 \pm 0.156$
Concrete	$6.624 \pm 0.31$	$10.446 \pm 0.432$	$8.713 \pm 0.748$	$10.445 \pm 0.438$
Energy Cooling	$1.692 \pm 0.158$	$3.215 \pm 0.1$	$2.497 \pm 0.173$	$3.215 \pm 0.099$
Energy Heating	$0.687 \pm 0.129$	$2.942 \pm 0.167$	$2.233 \pm 0.209$	$2.942 \pm 0.168$
Tower Data	$29.295 \pm 2.043$	$30.453 \pm 3.247$	$14.064 \pm 0.5$	$30.432 \pm 3.274$
Wine Red	$0.676 \pm 0.255$	$0.652 \pm 0.025$	$0.686 \pm 0.033$	$0.652 \pm 0.025$
Wine White	$0.745 \pm 0.052$	$0.754 \pm 0.012$	$0.725 \pm 0.018$	$0.754 \pm 0.012$
Yacht	$0.904 \pm 0.691$	$8.997 \pm 0.616$	$9.711 \pm 1.47$	$9.012 \pm 0.606$
Pontuação geral	2.5	4.375	3.25	3.75
Classificação	1	6	3	5
Base de dados	MLP	OLS	Ridge	XGBoost
Airfoil	$6.983 \pm 0.28$	$4.819 \pm 0.156$	$4.819 \pm 0.156$	$2.021 \pm 0.151$
Concrete	$16.764 \pm 0.623$	$10.445 \pm 0.438$	$10.445 \pm 0.438$	$5.976 \pm 0.769$
Energy Cooling	$12.194 \pm 0.894$	$3.211 \pm 0.1$	$3.215 \pm 0.099$	$0.661 \pm 0.066$
Energy Heating	$13.355 \pm 0.428$	$2.935 \pm 0.164$	$2.942 \pm 0.168$	$0.317 \pm 0.029$
Tower Data	$87.76 \pm 0.764$	$30.432 \pm 3.274$	$30.432 \pm 3.274$	$25.207 \pm 1.326$
Wine Red	$0.952 \pm 0.024$	$0.652 \pm 0.025$	$0.652 \pm 0.025$	$0.756 \pm 0.038$
Wine White	$1.016 \pm 0.018$	$0.754 \pm 0.012$	$0.754 \pm 0.012$	$0.816 \pm 0.025$
Yacht	$15.632 \pm 0.811$	$9.012 \pm 0.606$	$9.012 \pm 0.606$	$1.094 \pm 0.284$
Pontuação geral	8.0	3.5	3.75	2.75
classificação	8	4	5	2

# Desempenho dos algoritmos comparados

- Apesar do bom desempenho do método **XGBoost** na metade das bases de dados, nos demais outros ele fica em colocações muito mais baixas que o IT-MUT, por isso mesmo sendo o primeiro em vários ele ainda é o segundo na classificação geral;
- Os métodos de regressão linear LARS, OLS e Ridge apresentaram um bom resultado entre eles, sendo que o LARS e o Ridge empataram na pontuação geral;
- Comparado com todas as outras técnicas, o melhor algoritmo proposto neste trabalho obteve também o melhor desempenho entre elas.

# Desempenho dos algoritmos comparados

Dataset	IT-MUT	Bayesian	LARS lasso	MLP
Airfoil	$2.758 \pm 0.433$	$4.819 \pm 0.156$	$4.819 \pm 0.156$	$8.893 \pm 0.482$
Concrete	$6.624 \pm 0.31$	$10.446 \pm 0.432$	$10.447 \pm 0.447$	$22.128 \pm 1.398$
EnergyCooling	$1.692 \pm 0.158$	$3.215 \pm 0.1$	$3.215 \pm 0.098$	$13.212 \pm 0.599$
EnergyHeating	$0.687 \pm 0.129$	$2.942 \pm 0.167$	$2.942 \pm 0.168$	$14.122 \pm 0.241$
TowerData	$29.295 \pm 2.043$	$30.666 \pm 3.141$	$30.44 \pm 3.271$	$120.821 \pm 7.731$
WineRed	$0.676 \pm 0.255$	$0.651 \pm 0.026$	$0.651 \pm 0.026$	$0.948 \pm 0.037$
WineWhite	$0.745 \pm 0.052$	$0.754 \pm 0.011$	$0.756 \pm 0.011$	$1.064 \pm 0.065$
Yacht	$0.904 \pm 0.691$	$8.998 \pm 0.622$	$8.963 \pm 0.612$	$20.757 \pm 1.656$
Pontuação geral	3.0	4.75	4.75	8.0
classificação geral	3	5	5	8
Dataset	OLS	Ridge	XGBoost	KNN
Airfoil	$4.819 \pm 0.156$	$4.82 \pm 0.156$	$1.612 \pm 0.131$	$2.684 \pm 0.146$
Concrete	$10.445 \pm 0.438$	$10.447 \pm 0.442$	$4.3 \pm 0.285$	$8.713 \pm 0.748$
EnergyCooling	$3.211 \pm 0.1$	$3.215 \pm 0.099$	$0.857 \pm 0.139$	$2.531 \pm 0.054$
EnergyHeating	$2.935 \pm 0.164$	$2.942 \pm 0.168$	$0.356 \pm 0.049$	$2.281 \pm 0.147$
TowerData	$30.432 \pm 3.274$	$30.542 \pm 3.223$	$15.665 \pm 0.698$	$14.064 \pm 0.5$
WineRed	$0.652 \pm 0.025$	$0.651 \pm 0.025$	$0.613 \pm 0.036$	$0.657 \pm 0.037$
WineWhite	$0.754 \pm 0.012$	$0.754 \pm 0.011$	$0.667 \pm 0.006$	$0.719 \pm 0.012$
Yacht	$9.012 \pm 0.606$	$8.996 \pm 0.626$	$0.672 \pm 0.086$	$8.952 \pm 1.622$
Pontuação geral	4.5	5.0	1.125	2.875
classificação geral	4	7	1	2

# Desempenho dos algoritmos comparados

- 1º - **XGBoost**. A tendência do XGBoost é apresentar um erro cada vez menor quanto mais estimadores.
- 2º - **KNN**. Método baseado em instância, que armazena toda a informação disponível para treino e, no momento de fazer uma nova predição, utiliza todos os dados de treino para estimar o valor de  $y$ .
- 3º - **IT-MUT**, próximo do 2º, distante do 1º e 4º. Nenhum método paramétrico superou o IT-MUT.
- A vantagem do IT-MUT está no resultado que ele fornece: uma função simbólica — ao contrário de um conjunto grande de árvores complexas ou o uso de informações dos vizinhos próximos.



# Índice

## 7 Conclusão

### Conclusão final

# Conclusão

- Aprofundamento dos estudos da aplicação da estrutura IT para regressão simbólica, entregando três algoritmos evolutivos;
- O IT-MUT se mostrou boa na minimização do erro de predição; o IT-CX em minimização da complexidade dos resultados; e o ITEA apresentou um ponto intermediário entre as duas.

# Conclusão

- Os resultados são uma prova de que é possível realizar uma regressão simbólica em um espaço de busca restringido e ainda sim obter um desempenho competitivo, com uma maior interpretabilidade devido à maior simplicidade resultante da representação IT.
- Os três algoritmos apresentaram uma convergência rápida, levando cerca de 5 minutos para evoluir uma população pequena por poucas gerações e obter um resultado relevante.

# Considerações finais

Apesar da limitação da estrutura, a não linearidade inserida pelos expoentes e pela função de transformação dá maior expressividade para os algoritmos, podendo não encontrar a equação que deu origem aos dados, mas uma que se aproxime de forma satisfatória — mas ainda existe espaço para muitas outras otimizações e ideias conceituais que possam melhorar o desempenho, a interpretabilidade e o tempo de execução.