Universidade Federal de Minas Gerais
Disciplina de Computação Natural
Prof Gisele L. Pappa

Trabalho Prático 1
Programação Genética

Data de Entrega: 08 de maio de 2023

---

1 Introdução

O principal objetivo deste trabalho é desenvolver conceitos chave para a construção de soluções para problemas usando Programação Genética (GP), envolvendo o entendimento e a implementação dos componentes básicos de um arcabouço de GP, bem como a análise de sensibilidade dos seus parâmetros (como eles afetam o resultado final, a natureza da convergência, etc) e procedimentos para avaliação das soluções alcançadas.

Uma dos problemas mais populares que podem ser resolvidos com técnicas de programação genética é a regressão simbólica. Conforme visto em sala de aula, dado um conjunto de m amostras provenientes de uma função *desconhecida* $f : \mathbb{R}^n \mapsto \mathbb{R}$, representadas por uma dupla $\langle X, Y \rangle$ onde $X \in \mathbb{R}^{m \times n}$ e $Y \in \mathbb{R}^m$, o objetivo é encontrar a expressão simbólica de f que melhor se ajusta às amostras fornecidas.

No arcabouço de programação genética a ser desenvolvido, os indivíduos deverão ser representados por árvores, compostas por nós terminais e operadores. Será de sua responsabilidade determinar ambos os conjuntos para solucionar o problema de regressão simbólica fornecido. Lembre-se que é importante considerar a presença de constantes (para a representação de coeficientes), bem como das variáveis do problema.

Um critério de avaliação possível para medir a qualidade de um indivíduo é a raiz quadrada do erro quadrático médio (RMSE)

$$f(Ind) = \sqrt{\frac{1}{N}\sum_{i=1}^{N} (\textsc{Eval}(Ind, x) - y)^2}$$

onde *Ind* é o indivíduo sendo avaliado, Eval(*Ind*, x) avalia o indivíduo *Ind* no conjunto de entrada fornecido x, y é a saída correta da função para a entrada x, e N é o número de exemplos fornecidos.

Decisões de Implementação:

1. Como representar um indivíduo (genótipo);
2. Como gerar a população inicial;
3. Quais operadores genéticos serão utilizados;
4. Facilidades para variação de parâmetros—parâmetros *hardcoded* no arcabouço certamente dificultarão a avaliação dos parâmetros;

2 Estudos do impacto do mecanismos de seleção

Um dos focos desse trabalho será na análise do impacto de diferentes métodos de seleção nos resultados alcançados. A princípio, testaremos as seleções por roleta e seleção por torneio, conforme visto em sala de aula, e compararemos essas abordagens com uma forma de seleção considerada por muitos estado da arte atualmente, chamada seleção *lexicase*.

A ideia da seleção lexicase é que, em cada processo de seleção, os indivíduos sejam avaliados em uma sequência aleatória de exemplos de treinamento, que chamaremos de casos, e apenas indivíduos cujo erro seja mínimo entre todos os casos (exemplos) considerados sobrevivam. Ela funciona da seguinte forma:

1. Todos os indivíduos da população são considerados candidatos para seleção;
2. Os exemplos de treinamento são embaralhados;
3. Indivíduos candidatos são avaliados para o primeiro caso, e aqueles com fitness pior que a melhor fitness para esse caso são removidos do conjunto de candidatos a pais;
4. Se houver mais de um indivíduo no conjunto de candidatos, o caso atual é removido do conjunto de exemplos e o passo 3 é repetido com o próximo exemplo (caso). Se houver apenas um indivíduo no conjunto de candidatos, ele é selecionado como pai. Se não houverem mais exemplos para serem avaliados, escolhe-se um indivíduo aleatoriamente do conjunto de candidatos.

Dessa forma, nem sempre todos os exemplos de treinamento são avaliados. Em sua definição clássica, para que o indivíduo i passe pelo caso de teste t, ele deve apresentar um erro mínimo $e_t^{(i)} = e_t^*$, onde e_t^* é o menor erro da população p no caso de teste t. Para problemas de regressão, uma adaptação desse mecanismo, chamada ϵ-lexicase, usa um parâmetro ϵ para definir um intervalo ao qual o erro do indivíduo naquele caso de teste deve pertencer para passar pelo caso de teste, ao invés de utilizar um valor exato.

Assim, definimos *epsilon* – $\epsilon_{e\lambda}$ – como um limiar que dita a condição que o indivíduo i tem que obedecer para passar o teste t, definida como $p_t(i)$:

$$\epsilon_{e\lambda} : p_t(i) = I(e_t(i) < e_t^* + \lambda(e_t)) \tag{1}$$

onde I é uma função indicadora que retorna 1 se verdadeiro e 0 se falso. Na Eq. 1, ϵ_e define $p_t(i)$ relativo a e_t^*, e por isso sempre ao menos um indivíduo de P passa pelo caso de teste. Embora ϵ possa ser uma constante definida pelo usuário, aqui ela será definida de acordo com a mediana do desvio absoluto (MAD) de e_t considerando todos os indivíduos candidatos, definida na Eq. 1 como λ:

$$MAD(e_t) = \lambda(e_t) = median_j(|e_t^j - median_k(e_t^k)|) \tag{2}$$

3 Bases de Dados

Três conjuntos de dados serão utilizados neste trabalho, e estão todos disponíveis no Moodle. Cada base de dados possui dois arquivos:

1. <nome-da-base>-train.csv

Tabela 1: Bases de dados disponibilizadas.

Base	**# atributos**	**# instâncias**		**Tipo**
		Treino	**Teste**	
synth1	3	60	600	Sintética
synth2	3	300	1000	Sintética
concrete[1]	9	824	206	Real

[1] Encontrada em `http://archive.ics.uci.edu/ml/datasets/concrete+compressive+strength`.

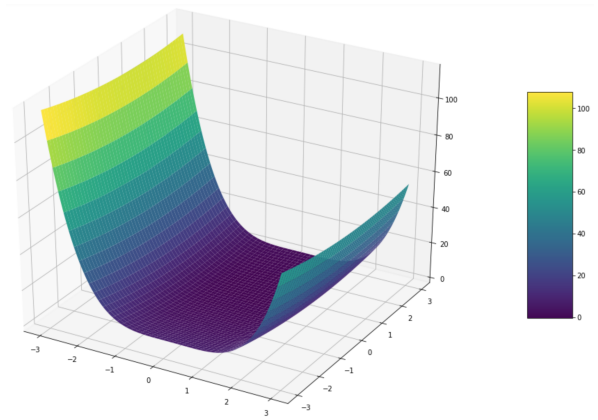

Figura 1: Base de dados synth1.

2. <nome-da-base>-test.csv

O primeiro deverá ser usado para evoluir às soluções até o número máximo de gerações ser alcançado. Finalizada esta etapa, as melhores soluções encontradas deverão ser avaliadas utilizando a base de teste. Neste momento é importante comparar os indicadores de teste com os indicadores de treino afim de detectar algum tipo de anomalia (overfitting, por exemplo). Note que aqui nãop usaremos validação cruzada e faremos apenas comparações com diferentes execuções do algoritmo com sementes aleatórias. Essa decisão foi tomada para reduzir o tempo de experimentos do trabalho.

Todos os arquivos estão no formato CSV, e a última coluna contém a saída desejada (y). Esta saída deverá ser comparada com a saída estimada para gerar o erro e.

4 Metodologia Experimental

O GP deve ser testado nas 3 bases de dados descritas na Tabela 1. A avaliação experimental descrita abaixo deve ser feita para uma das bases representando problemas sintéticos e para o problema real. Os parâmetros considerados mais apropriados para o base sintética escolhida devem ser novamente utilizados para a outra bases sintética (correspondente aos problemas listados na Tabela 1).

A parte de escolha e estudo dos parâmetros deve ser feita da seguinte forma:

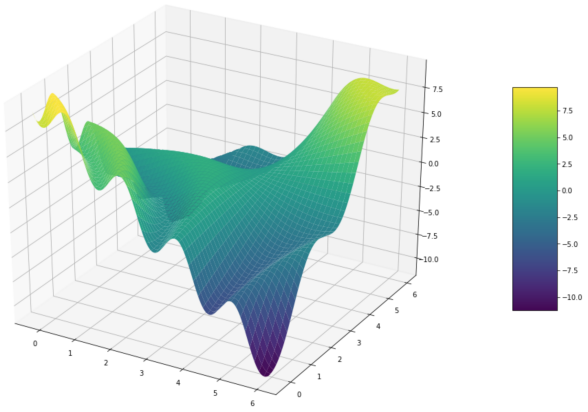

Figura 2: Base de dados synth2.

- Definir o tamanho máximo do indivíduo como 7. Esse parâmetro não precisa ser obrigatoriamente variado.
- Escolher o tamanho da população e o número de gerações apropriados. O tamanho da população pode ser testado, por exemplo, utilizando 50, 100, 500 indivíduos. O número de gerações pode também ser escolhido usando esses mesmos números. Mas como saber se o escolhido é o mais apropriado? Vocês podem avaliar como o aumento no número da população ou de gerações melhora a solução encontrada (em termos do erro gerado), se a população converge, etc.
- Testar duas configurações de parâmetros para crossover e mutação. Na primeira, a probabilidade de crossover (p_c) deve ser alta (por exemplo, 0.9), e a probabilidade de mutação (p_m) deve ser baixa (por exemplo, 0.05). Na segunda, p_c deve ser mais baixa (por exemplo, 0.6) e p_m mais alta (por exemplo, 0.3). Para ambas as configurações, deve-se avaliar o efeito do crossover e da mutação na evolução, isto é, em quantos casos esses operadores contribuem positivamente (os filhos gerados são melhores que os pais) ou negativamente para a evolução? A partir desse estudo inicial, que valores finais você proporia?
- Analisar as mudanças ocorridas quando se muda o método de seleção.
- Utilizar elitismo.
- Existe uma forma simples de medir bloating no seu algoritmo?

Lembrem-se que ao mexer em um dos parâmetros, todos os outros devem ser mantidos constantes, e que a análise dos parâmetros é de certa forma interativa. A configuração de parâmetros raramente vai ser ótima, mas pequenos testes podem melhorar a qualidade das soluções encontradas.

Por ser um método estocástico, a avaliação experimental do algoritmo baseado em GP deve ser realizada com *repetições*, de forma que os resultados possam ser reportados segundo o valor médio obtido e o respectivo desvio-padrão. A realização de 30 repetições pode ser um bom ponto de partida (lembrando que desvio-padrão alto sugere um maior número de repetições).

Guia sugerido para execução dos experimentos

1. Escolha o tamanho da população e número de gerações (utilizar tamanho máximo do indivíduo como 7, elitismo, torneio de tamanho 2 e $p_c = 0.9$ e $p_m = 0.05$).
2. Definidos o tamanho da população e número de gerações, troque o tipo de seleção.
3. Após alguns testes, escolha o método de seleção mais apropriado e varie p_c e depois p_m. Os parâmetros escolhidos no anteriormente ainda são apropriados?
4. Escolha os melhores parâmetros dos anteriores e retire o elitismo. Os resultados obtidos são os mesmos?
5. Se desejar, teste outras características, como métodos para garantir a diversidade da população.

Estatísticas importantes

Estas estatísticas devem ser coletadas para todas as gerações.

1. Fitness do melhor e pior indivíduos
2. Fitness média da população
3. Número de indivíduos repetidos na população
4. Número de indivíduos gerados por crossover melhores e piores que a fitness média dos pais

O que deve ser entregue...

- Código fonte do programa
- Documentação do trabalho:
  - Introdução
  - Implementação: descrição sobre a implementação do programa, incluindo detalhes da representação, fitness e operadores utilizados
  - Experimentos: Análise do impacto dos parâmetros no resultado obtido pelo AE.
  - Conclusões
  - Bibliografia

A entrega DEVE ser feita pelo Moodle na forma de um único arquivo zipado, contendo o código e a documentação do trabalho.

Considerações Finais

- Os parâmetros listados para execução dos experimentos são sugestões iniciais, e podem ser modificados a sua conveniência.
- Depois da entrega do trabalho, faremos uma competição em sala de aula para avaliar as diversas decisões de implementação do algoritmo e como a otimização dos parâmetros podem levar ao sucesso ou fracasso do algoritmo.

5 Referências

La Cava et al, ϵ-Lexicase selection for Regression, `https://arxiv.org/pdf/1905.13266.pdf`