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

Trabalho Prático 1
Programação Genética

Data de Entrega: 05 de outubro de 2015

---

1 Introdução

O principal objetivo deste trabalho é desenvolver conceitos chave para a construção de soluções para problemas usando Programação Genética (PG), envolvendo o entendimento e a implementação dos componentes básicos de um arcabouço de PG, 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, lembrando 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 soma do erro absoluto, dada por:

$$f(Ind) = \sum_{\{\langle x,y \rangle\}} \|\textsc{Eval}(Ind, x) - y\|,$$

onde *Ind* é o indivíduo sendo avaliado, Eval(*Ind*, x) avalia o indivíduo *Ind* com a variável x, $\{\langle x, y \rangle\}$ é o conjunto de entrada fornecido e y é a saída correta da função para a entrada x.

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;
5. Como prover uma avaliação de fitness que seja facilmente instanciada para outros problemas (fenótipo)—por exemplo, deve ser fácil incluir funções para cálculo da fitness dos indivíduos de forma específica para um determinado problema.

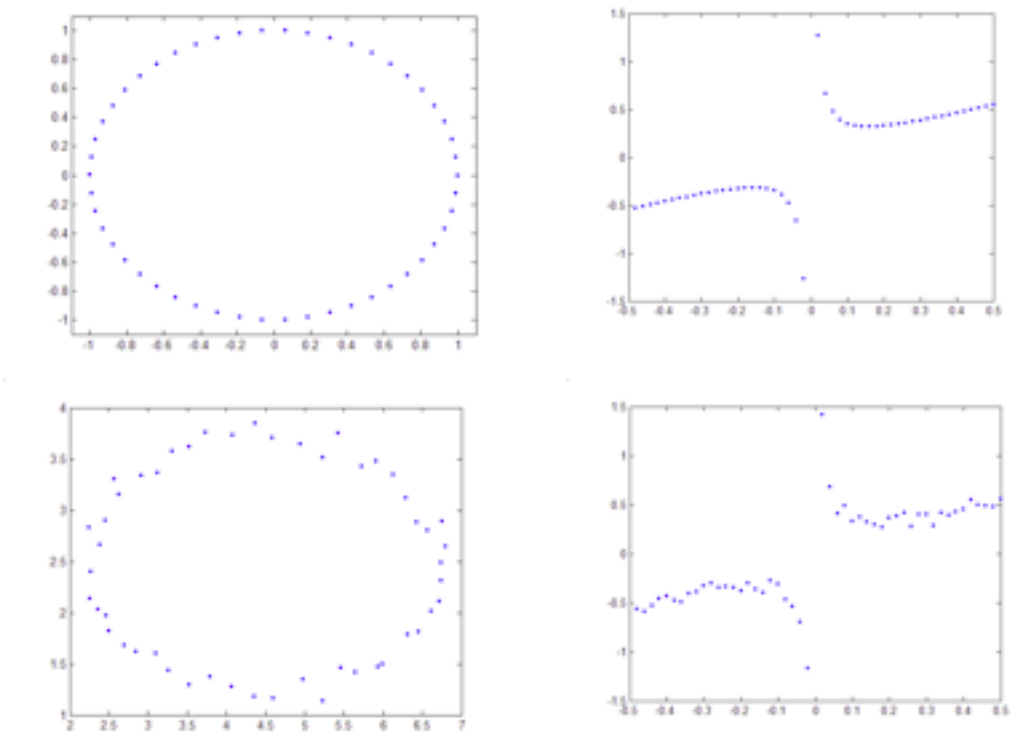

Figura 1: Exemplos das 4 primeiras funções que devem ser encontradas.

2 Bases de Dados

Cinco conjuntos de dados serão utilizados neste trabalho, e estão todos disponíveis no Moodle. Esses cinco conjunto representam na verdade 3 problemas diferentes. Os dois primeiros problemas tratam de funções conhecidas para testes de sanidade do algoritmo implementado, e são mostradas na Figura 1. Nesse caso, os algoritmos devem ser exectuados considerando os 50 pontos $< x, y >$ dados como entrada. Note que, do lado esquerdo da figura, temos uma elipse perfeita e abaixo a mesma elipse com perturbações nos dados de entrada. O mesmo vale para o problema apresentado do lado direito da figura. O principal objetivo é entender como o algoritmo se comportará com a introduação de ruído nos dados. Note que essas quatro bases de dados possuem apenas uma variável de entrada x e sua respectiva saída y.

Após esses teste, lidaremos com um problema real, disponível no repositório de problemas de aprendizado de máquina da UCI, e conhecido comom o *Yacht Hydrodynamics Data Set*[1]. Nesse caso, temos 6 variáveis de entrada e uma variável de saída, que devem ser ajustadas considerando 308 pontos.

3 Metodologia Experimental

O GP deve ser testado nas 5 bases de dados descritas acima. A avaliação experimental descrita abaixo deve ser feita para uma das bases representando problemas simples (justifique sua escolha) e para o problema real. Os parâmetros considerados mais apropriados para o

[1] https://archive.ics.uci.edu/ml/datasets/Yacht+Hydrodynamics

base simples escolhida devem ser novamente utilizados para as outras bases simples (correspondentes aos problemas listados na Figura 1).

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

- 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 3 500 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 o tamanho do torneio aumenta de 2 para 5 ou 3 para 7, dependendo do tamanho inicial da população.
- 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 PG 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 para execução dos experimentos

1. Escolha do 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. Após alguns testes, defina o tamanho da população e o número de gerações e varie p_c e p_m. Os parâmetros escolhidos no passo 1 ainda são apropriados?
3. Definidos o tamanho da população, número de gerações, p_c e p_m, aumente o tamanho do torneio.

4. Escolha os melhores parâmetros dos anteriores e retire o elitismo. Os resultados obtidos são os mesmos?

5. Se desejar, teste outras características nesse problema.

Estatísticas importantes

Essas 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. O grupo vencedor receberá um prêmio (que poderá ser em nota, chocolate, etc).