

Complemento IV – Introdução aos Algoritmos Genéticos

Esse documento é parte integrante do material fornecido pela WEB para a 2ª edição do livro *Data Mining: Conceitos, técnicas, algoritmos, orientações e aplicações*.

Introdução

Algoritmos Genéticos são modelos computacionais de busca e otimização de soluções em problemas complexos, inspirados em princípios da teoria da evolução natural de Charles Darwin e da reprodução genética.

Segundo o princípio básico da evolução natural de Darwin, indivíduos mais aptos possuem maiores chances de sobrevivência e, conseqüentemente, mais oportunidades de gerarem descendentes e perpetuarem seus códigos genéticos pelas gerações seguintes. A identificação de cada indivíduo é expressa pelo seu código genético que fica representado nos cromossomas deste indivíduo.

Resumidamente, Algoritmos Genéticos são técnicas que procuram obter boas soluções para problemas complexos por meio da evolução de populações de soluções codificadas em cromossomas artificiais. Todo problema de difícil modelagem matemática ou com um número muito grande, possivelmente infinito, de soluções é considerado um problema complexo. A tabela abaixo apresenta a analogia entre Algoritmos Genéticos e a Evolução Natural.

Algoritmos Genéticos empregam um processo adaptativo e paralelo de busca de soluções em problemas complexos. O processo é adaptativo, pois as soluções existentes a cada instante influenciam a busca por futuras soluções. O paralelismo do processo é decorrência natural do fato de que diversas soluções são consideradas a cada momento pelos Algoritmos Genéticos. Exemplos de aplicações de Algoritmos Genéticos:

- Otimização de funções matemáticas
- Otimização Combinatorial
- Otimização de Planejamento
- Problema do Caixeiro Viajante
- Problema de Otimização de Rota de Veículos
- Otimização de Distribuição e Logística
- Seleção de Variáveis em Mineração de Dados

Nos Algoritmos Genéticos, um cromossoma é uma estrutura de dados que representa uma das possíveis soluções do espaço de busca de um problema. Cromossomas são então

submetidos a um processo evolucionário que avalia, seleciona e combina soluções ao longo de diversos ciclos, procurando obter indivíduos mais aptos (melhores soluções).

Tabela: Analogia entre Algoritmos Genéticos e a Natureza

Evolução Natural	Algoritmos Genéticos
Meio Ambiente	Problema
Indivíduo	Solução
Cromossoma	Representação (palavra binária, vetor, etc)
Gene	Característica do Problema
Alelo	Valor da Característica
Reprodução Sexual	Operador de Cruzamento
Mutação	Operador de Mutação
População	Conjunto de Soluções
Gerações	Ciclos

Em geral, um Algoritmo Genético é caracterizado pelos seguintes componentes:

- Problema
- Representação das Soluções do Problema
- Decodificação do Cromossoma
- Avaliação
- Seleção
- Operadores Genéticos
- Técnicas e Parâmetros

Abaixo seguem alguns comentários relevantes acerca de cada um destes componentes.

Problema

Algoritmos Genéticos são particularmente úteis na solução de problemas complexos que envolvem otimização. Estes são problemas com diversos parâmetros ou características que precisam ser combinadas em busca de soluções melhores. Possuem, em geral, muitas restrições aos valores que os parâmetros podem ter, além de grandes espaços de busca.

Considere o problema de encontrar o valor máximo da função $f(x)=x^2$, sendo x um valor inteiro no intervalo $[0; 63]$. Evidentemente este não é um problema complexo, pois o espaço de busca compreende apenas 64 pontos inteiros a serem avaliados. Além disso, um conhecimento prévio em Análise Matemática, nos permite facilmente concluir que o ponto máximo desta função é $(63; 3969)$. No entanto, este exemplo será utilizado para fins didáticos na ilustração dos demais conceitos envolvidos na utilização de Algoritmos Genéticos.

Representação das Soluções do Problema

A menor unidade de um Algoritmo Genético é chamada gene. Um gene representa uma unidade de informação do domínio do problema. Uma série de genes forma um cromossoma, que representa uma possível solução completa para o problema.

A representação de possíveis soluções do espaço de busca de um problema define a estrutura do cromossoma a ser manipulado pelo algoritmo. Qualquer que seja a forma de representação de um problema, esta deve ser capaz de representar todas as soluções do espaço de busca que se deseja investigar. A representação binária é a mais empregada, devido à sua simplicidade e facilidade de manipulação.

Para o problema acima, é necessária uma representação para números inteiros. Podem ser utilizados binários representando números inteiros. Neste caso, seriam necessários cromossomas de 6 bits para representar números entre 0 e 63. Exemplos:

C1: 001001 representa $x=9$

C2:000100 representa $x=4$

Há problemas em que valores binários podem ser utilizados para representar números reais com uma precisão de p casas decimais. Supondo que os números reais pertençam a um intervalo $[X_{min}, X_{max}]$, são necessários k bits de forma que k atenda à seguinte relação:

$$2^k \leq (X_{max} - X_{min}) * 10^p$$

Nem sempre a representação binária pode ser empregada. Em muitos casos, o problema requer um alfabeto de representações com mais símbolos.

Decodificação do Cromossoma

O processo de decodificação de um cromossoma consiste na construção da solução a ser avaliada pelo problema. No exemplo acima, a decodificação consiste em transformar a string binária em um número inteiro.

Avaliação

Para que um cromossoma seja avaliado é necessário converter o cromossoma numa solução para o problema, isto é, decodificar o cromossoma.

A avaliação de cada solução em um Algoritmo Genético é um processo realizado por uma função que tem por objetivo fornecer uma medida da qualidade de tal solução (aptidão) frente ao problema.

As funções de avaliação variam de um problema para outro. No exemplo acima, a função $f(x)=x^2$ expressa a qualidade de cada indivíduo como solução para o problema. $C1$ é um indivíduo melhor (mais apto) que $C2$, pois $9^2 > 4^2$.

Seleção

Durante o processo de seleção, o Algoritmo Genético escolhe os cromossomas, privilegiando aqueles com maiores aptidões para permanecer e se multiplicar na população. Uma forma comum de seleção é aquela em que cada cromossoma tem a probabilidade de permanecer na próxima geração proporcional à sua aptidão (método da roleta). Cromossomas com maiores aptidões possuem mais espaço na roleta e conseqüentemente possuem maiores chances de serem escolhidos para o processo de reprodução. Assim, se f_i é a avaliação do i -ésimo indivíduo na população corrente, a probabilidade p_i (aptidão relativa) deste indivíduo ser selecionado é proporcional a:

$$p_i = \frac{f_i}{\sum_{i=1}^n f_i}$$

Onde n é o número de indivíduos na população.

No exemplo anterior, supondo que existam apenas os indivíduos *C1* e *C2*:

Indivíduo	Aptidão Relativa
C1	$p_1=81/(81+16)=0.84$
C2	$p_2=16/(81+16)=0.16$

No método da roleta, os indivíduos são organizados em ordem crescente de aptidão, sendo sorteado um número entre 0 e 1. O indivíduo selecionado é o primeiro indivíduo cuja aptidão acumulada seja superior ao número sorteado. No exemplo:

Indivíduo	Aptidão Relativa Acumulada
C1	0.84
C2	1.00 (0.84 + 0.16)

Para números sorteados até 0.84, o indivíduo selecionado seria *C1*. Para valores acima de 0.84, o indivíduo selecionado seria *C2*.

Operadores Genéticos

Uma vez selecionados os indivíduos, o Algoritmo Genético cria novas soluções através da combinação e refinamento das informações dos cromossomas usando duas operações genéticas básicas: *crossover* e *mutação*. Essas operações produzem novas soluções que formam uma nova população. Cada nova população é chamada de geração.

Durante o *crossover*, dois cromossomas trocam algumas de suas informações contidas em determinados genes. Novos indivíduos são criados a partir da troca de material genético entre seus pais. Os descendentes possuem características genéticas de ambos os genitores. Por exemplo, sendo *C1* e *C2*, os genitores e 3 o ponto de corte (ou de cruzamento) aleatoriamente selecionado (operador de *crossover* de um ponto), *D1* e *D2* seriam os descendentes nesta situação.

C1: 001001

C2: 000100

D1: 001100

D2: 000001

D1 é um cromossoma mais apto que seus genitores. No entanto, *D2* tornou-se um indivíduo bem menos apto.

Os cromossomas criados a partir do operador de *crossover* são submetidos à operação de mutação. A mutação permite explorar o potencial de novos indivíduos, aumentando a diversidade de indivíduos na população. O operador de mutação altera o conteúdo de uma posição do cromossoma, com uma probabilidade, em geral, bem baixa (<1%). Exemplo do indivíduo *C2* antes e após sofrer mutação no último bit:

C2: 000100, antes da mutação

C2: 000101, após a mutação

Embora os operadores genéticos costumem variar em função do problema, três dos tipos de operadores de *crossover* mais comuns são:

- **Crossover de um ponto** - O *crossover* de um ponto recombina duas soluções a partir de um único ponto de corte aleatório. Este foi o operador utilizado no exemplo acima.
- **Crossover de dois pontos** - no *crossover* de dois são dois os pontos aleatórios escolhidos como referência para a troca de valor dos atributos (genes).

Operação de crossover com dois pontos de corte

Cromosso ma	Palavra			Nova Palavra
A	1	001	1	1001111
	0	0	0	0
C	1	011	1	1100101
	1	1	1	1

- **Crossover uniforme** – Este tipo de *crossover*, por sua vez, é capaz de recombinar quaisquer posições entre dois genitores. Para tanto, utiliza uma palavra binária escolhida aleatoriamente para designar os bits selecionados em cada genitor na criação dos descendentes. Exemplo:

G1:	1100101
G2:	0111110
Padrão:	0110100
D1:	0101110
D2:	1110101

Técnicas e Parâmetros

Em geral, as técnicas, os parâmetros e os tipos de operadores genéticos afetam significativamente o desempenho dos Algoritmos Genéticos. A escolha de técnicas, parâmetros e operadores é empírica, porém em sintonia com o tipo de problema envolvido.

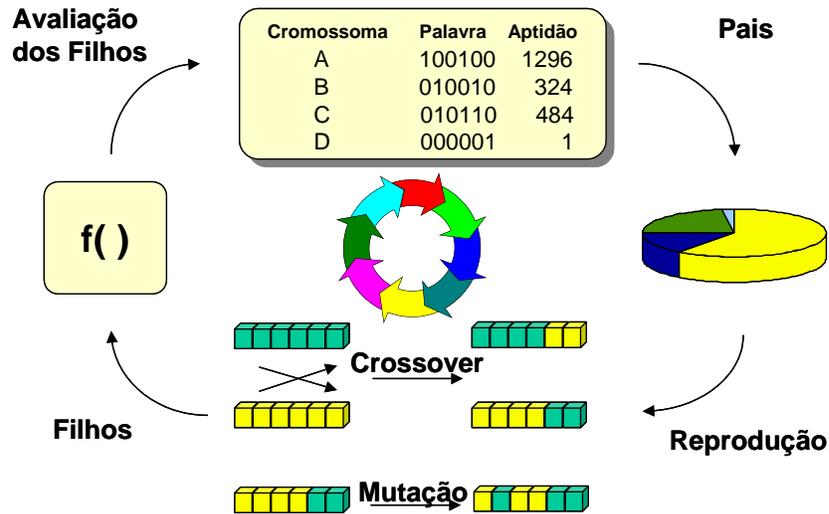
A inicialização da população determina o processo de criação dos indivíduos para o primeiro ciclo do algoritmo, sendo comum a formação da população inicial a partir de indivíduos aleatoriamente criados.

Em um Algoritmo Genético, vários parâmetros controlam a realização do processo evolucionário:

- **Tamanho da população** – números de pontos no espaço de busca sendo considerados em paralelo.
- **Taxa de Crossover** – probabilidade de um indivíduo ser recombinação geneticamente com outro.
- **Taxa de Mutação** - probabilidade do conteúdo de cada gene do cromossoma ser alterado.
- **Número de Gerações** – quantidade ciclos do Algoritmo Genético. Pode ser deduzido caso se tenha a informação do tamanho da população e total de indivíduos (total de indivíduos / tamanho da população).
- **Total de Indivíduos** – total de soluções a serem geradas e avaliadas pelo Algoritmo Genético (tamanho da população X número de gerações).

Os últimos dois parâmetros são normalmente empregados como critério de parada de um Algoritmo Genético, isto é, quando o processo evolucionário deve ser interrompido.

Um Algoritmo Genético pode ser descrito como um processo contínuo que repete ciclos de evolução controlados por um critério de parada, conforme ilustrado pela figura a seguir:



Ciclo de um Algoritmo Genético

As técnicas de reprodução determinam o critério de substituição dos indivíduos de uma população para a geração seguinte. A seguir encontram-se citados alguns exemplos:

- *Troca de toda a População*: A cada ciclo, N novos indivíduos são criados, substituindo a população anterior. Para tanto, $N/2$ pares são selecionados para acasalamento, gerando N descendentes.
- *Elitismo*: Todos os cromossomas são substituídos, sendo o cromossoma mais apto da população corrente copiado para a população seguinte.
- *Steady State*: Gera M indivíduos ($M < N$), que substituem os piores indivíduos da população corrente. Pode haver duplicação de indivíduos na população resultante.
- *Steady State sem Duplicados*: Similar à técnica de Steady State, não permitindo a presença de indivíduos duplicados na população.

As técnicas de aptidão influenciam na forma com que as aptidões dos cromossomas em uma população são numericamente atribuídas. A seguir encontram-se citados alguns exemplos:

- Método Clássico: Atribuir como aptidão de um cromossoma o valor numérico do resultado da avaliação. Este método, embora utilizado em muitos problemas, pode apresentar duas situações que precisam ser tratadas:
 - a) competição próxima – Indivíduos cujas aptidões são numericamente muito próximas, dificultando a distinção entre eles quanto à qualidade da solução proporcionada;
 - b) super-indivíduos – São indivíduos com avaliação muito superior à média, que podem dominar o processo de seleção, impedindo que o Algoritmo Genético obtenha novas soluções, potencialmente melhores do que as representadas pelos super-indivíduos.
- Normalização Linear: Consiste em ordenar os cromossomas em ordem crescente de avaliação e atribuir linearmente (equação abaixo) valores de aptidões aos cromossomas entre um intervalo [valor mínimo, valor máximo], distanciados de um valor fixo entre eles.

$$A_i = \min + \left[\frac{\max - \min}{N - 1} \right] * (i - 1)$$

As técnicas de interpolação de parâmetros de um Algoritmo Genético têm como objetivo buscar o valor ideal de um determinado parâmetro para cada ciclo, durante toda a evolução.

Pode-se intuitivamente perceber que a taxa de aplicação do operador de *crossover* deve ser maior nas primeiras gerações, quando a população se apresenta dispersa pelo espaço de busca. Após várias gerações, os indivíduos tendem a apresentar, na sua maioria, características muito similares. Um incremento da taxa de mutação nesta fase da evolução propicia uma pequena dispersão da população, trazendo novo material genético para a formação de melhores indivíduos. Pelo exposto acima, pode-se perceber ainda a necessidade de que parâmetros do Algoritmo Genético sejam interpolados ao longo do processo evolutivo. Em linhas gerais, a interpolação de parâmetros pode ser linear ou adaptativa.

Na interpolação linear, uma taxa ou um parâmetro é variado entre um valor inicial e um valor final, por meio de incrementos/decrementos fixos a cada k gerações. A interpolação adaptativa, normalmente utilizada para ajuste da taxa de aplicação de operadores genéticos, considera o desempenho destes operadores nos ciclos anteriores. Este

desempenho é medido em função do sucesso dos referidos operadores na criação de melhores indivíduos.

Estrutura Geral de um Algoritmo Genético

A seguir se encontra a estrutura geral básica de um Algoritmo Genético. Esta estrutura está descrita em pseudo-código, pois tem como objetivo apresentar em nível lógico o processo evolutivo em Algoritmos Genéticos. Algumas alterações nesta estrutura são admissíveis em função do problema analisado. Por exemplo: podem ser incluídas funções para implementar técnicas de reprodução ou ainda técnicas de aptidão. Considerando a importância da função de avaliação, optou-se por apresentar em seguida a sua descrição básica. As demais funções não foram descritas, mas podem ser obtidas na maioria das aplicações de Algoritmos Genéticos disponíveis na Internet.

Diferentes critérios de parada podem ser utilizados para terminar a execução de um Algoritmo Genético. Exemplos: a) Após um determinado número de iterações (ciclos ou gerações); b) quando a aptidão média ou do melhor indivíduo não melhorar mais; c) quando as aptidões dos indivíduos de uma população se tornarem muito parecidas.

Estrutura Geral de um Algoritmo Genético

$T=0$

Gerar População Inicial $P(0)$

Avaliar $P(0)$

Enquanto Critério de Parada não for satisfeito **faça**:

- a. $T=T+1$
- b. Selecionar População $P(T)$ a partir de $P(T-1)$
- c. Aplicar Operadores de Cruzamento sobre $P(T)$
- d. Aplicar Operadores de Mutação sobre $P(T)$
- e. Avaliar $P(T)$

Fim Enquanto

Estrutura Geral de uma Função de Avaliação (Avaliar $P(T)$)

Para todo indivíduo i da População Atual $P(T)$ **faça**

Avaliar a aptidão do indivíduo i

Fim Para

Para Saber Mais

Para maiores detalhes sobre Algoritmos Genéticos, o leitor poderá recorrer às referências a seguir.

Linden, Ricardo. Algoritmos Genéticos. 3ª ed, Rio de Janeiro: Ciência Moderna, 2012, ISBN: 9788539901951.

Davis, L. Handbook of Genetic Algorithms. Thonson Comp. Press, 1990.

Michalewicz Genetic Algorithms + Data Structures = Evolution Programs. Springer-Verlag, 1994.