Florence Alyssa Sakuma Shibata, Shayenne da Luz Moura

Método Simplex
Fase 2

São Paulo

2015

Resumo

Este relatório trata do desenvolvimento do algoritmo simplex, com implementação revisada, que encontra uma solução ótima de problemas que possuam pelo menos uma solução viável básica e nenhuma delas seja degenerada, recebendo em sua inicialização uma solução viável básica. Por fim, todos os resultados foram condizentes com o esperado.

Palavras-chaves: solução ótima. simplex revisado. problema programação linear.

Sumário

Introdução

Em investigações científicas é comum buscar relações entre dados pontuais. Normalmente tem-se muitos pontos coletados em experimento e há sempre uma razão teórica para acreditar que esses pontos são aproximados por uma função.

Como é praticamente impossível que uma função atravesse exatamente todos os pontos de dados é interessante encontrar uma função que tenha a uma distância aceitável, ou seja, que possua o menor desvio médio possível.

Encontrar essa função é, na realidade, encontrar a solução de um sistema linear sobredeterminado, isto é, um sistema com mais equações que incógnitas, $Ax = b$, onde $A \in \mathbb{R}^{n \times m}, n \geq m \ e \ b \in \mathbb{R}^n$. Sendo A a matriz que define o sistema, n o número de equações, m o número de incógnitas e b os dados do problema.

Para analisar quão boa é essa aproximação utilizou-se o método dos quadrados mínimos. Para tal, deve-se encontrar a função que minimiza as distâncias entre a solução calculada e os valores observados.

Neste caso, o problema consiste em encontrar $x \in \mathbb{R}^n$ para o qual $\|r\|_2$ é minimizada, onde $r = b - Ax$ é o vetor de resíduos.

Para resolver esse problema, utilizou-se o conceito de matriz ortogonal, que é uma matriz quadrada cuja inversa coincide com sua transposta, por esta possuir número de condição igual a 1, isto é, é bem condicionada. Matrizes ortogonais nos permitem realizar operações com o sistema $Ax = b$ sem perder a precisão de sua solução.

Execução

Para compilar o programa basta digitar o comando:

`$ make` Para executá-lo é necessário passar o nome do arquivo com os dados da matriz. Por exemplo:

`$ ./main m1.txt`

Observação: A entrada da matriz possui o mesmo formato de entrada do EP1, porém na segunda linha haverá um inteiro m, indicando a dimensão da coluna da matriz.

1 Teoria do algoritmo

1.1 Desenvolvimento do algoritmo

1.2 Funcionamento do algoritmo

1.3 Complexidade

2 Testes

2.1 Solução ótima dada

2.2 Custo ótimo ilimitado

2.3 Solução ótima diferente

3 Resultados

4 Resolução de sistema com posto incompleto

Para o teste do posto incompleto, foi gerado $Ax = b$ onde:

$$A = \begin{pmatrix} 1.0000000 & 0.0000000 & 1.0000000 \\ 2.0000000 & 5.0000000 & 2.0000000 \\ -1.0000000 & -5.0000000 & -1.0000000 \\ -1.0000000 & 0.0000000 & -1.0000000 \end{pmatrix}$$

$$b = \begin{pmatrix} 10.000 \\ 45.000 \\ -35.000 \\ -10.000 \end{pmatrix}$$

com valores pré-determinados

$$x = \begin{pmatrix} 6 \\ 5 \\ 4 \end{pmatrix}$$

Note que a terceira e a quarta linha sao combinações lineares da primeira e segunda linha. Enviando a matriz A e o vetor b para o programa (arquivo postoinc.txt), obtemos o resultado $x_0 = 10$ e $x_1 = 5$, com resíduo 16.44817.

A solução é razoável, uma vez que a matriz A é singular, porém não é a melhor solução gerada, pois a encontrada pelo programa apresenta o resíduo de 16.448, enquanto o resíduo com o sistema utilizando x_1 é zero.

5 Resolução de sistema com posto completo e problema dos quadrados mínimos

Inicialmente foi tomado o polinômio $x^5 - 7x^4 + 5x^3 - 3x^2 + 2x - 1$ e sorteados 70 pontos para servirem como dados de entrada e foram calculadas as funções que melhor aproximariam esses pontos. As funções soluções foram encontradas em diferentes graus.

Após, os dados foram perturbados em 15% e foram calculadas novas soluções. Estas funçoes e suas características de aproximação estão descritas a seguir.

5.1 Dados iniciais

Os pontos sorteados e a função caracterizadas pelo polinômio

$$x^5 \;-\; 7x^4 \;+\; 5x^3 \;-\; 3x^2 \;+\; 2x \;-\; 1$$

estão plotados abaixo.

Os pontos foram usados para buscar funções que os atravessassem reduzindo o resíduo. A partir dessas soluções, os dados foram perturbados e suas novas funções calculadas.

5.2 Solução 2º grau

Ao buscar uma função de grau 2 para aproximar os pontos encontrou-se os seguintes coeficientes:

$x_0 = -3.74052$ e $x_1 = 5.87079$

Ao perturbar os dados de entrada em 15%, o programa calculou as funções com resíduos 80.79981 para perturbação em b e 88.55303 para perturbação em A e b.

5.3 Solução 3º grau

Ao buscar uma função mônica de grau 3 para aproximar os pontos encontrou-se os seguintes coeficientes:

$x_0 = -0.39367,\ x_1 = 5.45203$ e $x_2 = -9.03908$

Ao perturbar os dados de entrada em 15%, o programa calculou as funções com resíduos 69.37743 para perturbação em b e 68.57657 para perturbação em A e b.

5.4 Solução 4° grau

Ao buscar uma função mônica de grau 4 para aproximar os pontos encontrou-se os seguintes coeficientes: $x_0 = -0.34728,\ x_1 = 1.25949,\ x_2 = -8.95480$ e $x_3 = 6.95427$ Ao perturbar os dados de entrada em 15%, o programa calculou as funções com resíduos 64.67217 para perturbação em b e 56.54760 para perturbação em A e b.

5.5 Solução 5° grau

Ao buscar uma função mônica de grau 5 para aproximar os pontos encontrou-se os seguintes coeficientes:

$$x_0 = -0.99223,\ x_1 = 1.73653,\ x_2 = -2.98373,\ x_3 = 6.13233 \text{ e } x_4 = -7.05190$$

Ao perturbar os dados de entrada em 15%, o programa calculou as funções com resíduos 61.44919 para perturbação em b e 47.96817 para perturbação em A e b.

Note que a o resíduo ($||s||$) aumenta cada vez que o grau da solução do polinômio diminui. Isto indica que o 'erro' da aproximação está sendo cada vez maior.[1]

[1] Observação: estes testes podem ser realizados com os arquivos matgrau5.txt, matgrau4.txt, matgrau3.txt, matgrau2.txt.

Conclusão

O algoritmo simplex implementado para resolver problemas de programação linear com soluções viáveis básicas não degeneradas e com pelo menos uma ótima é correto, porém não é o mais eficiente.

Como descrito no detalhamento de sua complexidade, é possível modificá-lo de maneira que se mantenha correto e gaste tempo em ordem $O(nm^2)$.

Mantendo as hipóteses em todos os casos, obtêm-se os resultados esperados em problemas cuja solução ótima é dada, quando o custo ótimo é ilimitado, ou é necessário encontrar uma direção viável cujo custo seja ótimo.