MAC0300 - Métodos Numéricos da Álgebra Linear

Professor Walter Mascarenhas

3º EP - Compressão de imagens usando SVD

Data de Entrega: 08/12/2012

1 Objetivo

O objetivo deste EP é demonstrar o uso da Decomposição em Valores Singulares (do inglês, *Singular Value Decomposition* - SVD) em compressão de imagens coloridas (RGB). O método de compressão e descompressão que vamos usar não é necessariamente o melhor método, mas ele é interessante por dar uma interpretação à decomposição de uma matriz em valores singulares, mostrando que os maiores valores singulares tendem a corresponder às informações mais importantes.

A implementação da decomposição em valores singulares a ser feita fará uso de dois algoritmos importantes: Bidiagonalização de Golub-Kahan (redução à forma bidiagonal de uma matriz usando Householder) e Decomposição em Valores Singulares de Golub–Reinsch, um processo muito parecido com o algoritmo QR.

2 Especificações

2.0.1 Sistema Operacional

O programa deve compilar e funcionar no Linux, devendo ser executado pela linha de comando.

2.0.2 Linguagem de Programação

A implementação do programa pode ser realizada na linguagem a sua escolha, sendo recomendadas as linguagens: Octave, R e Python.

2.0.3 Parâmetros de Entrada

O programa deve receber por parâmetros da linha de comando o valor do rank k da compressão e o nome da imagem a ser compactada.

O nome do arquivo de imagem será da forma "nome.bmp". Exemplo: lena.bmp.

No exemplo, a execução de um programa em Octave, por um terminal, seria:
$ octave -k 128 lena.bmp

2.0.4 Arquivo de Saída

O programa deve salvar, na pasta de execução do programa, o arquivo da imagem compactada em formato bmp, de nome igual ao arquivo de entrada, seguido do sufixo "compressed". Exemplo:

lena-compressed.bmp

3 Métodos

3.1 Singular Value Decomposition

Considerando apenas os casos de matrizes com entradas reais, a decomposição em valores singulares de uma matriz M de dimensão $m \times n$ será:

$$M = U \Sigma V^T,$$

Onde:
U é uma matriz ortogonal ($U^T = U^{-1}$) de dimensão $m \times m$;
Σ é uma matriz retangular diagonal de dimensão $m \times n$, ou seja, $\Sigma = diag(\sigma_1, \sigma_2, \ldots, \sigma_p)$, com $p = \min{(m,n)}$ e $\sigma_1 \geq \sigma_2 \geq \cdots \geq 0$;
V é outra matriz ortogonal de dimensão $n \times n$.

Para efeito de nomenclatura, as entradas da diagonal $\Sigma(i,i) = \sigma_i$ serão chamados valores singulares de M. As m colunas de U e as n colunas de V serão chamados vetores singulares à esquerda e vetores singulares à direita de M, respectivamente.

Os valores singulares são as raízes quadradas dos autovalores de MM^T ou M^TM, desconsiderando aqueles muitos próximos de zero. Dessa forma, para encontrar a decomposição de M em valores singulares, precisamos encontrar os autovalores e os autovetores de MM^T e M^TM. Os autovetores de M^TM formam as colunas de V, os autovetores de MM^T formam as colunas de U e as raízes quadradas dos autovalores de MM^T ou M^TM, rearranjados na ordem decrescente, são as entradas da diagonal de Σ.

Isso pode ser demonstrado como abaixo:
Dado que $M = U\Sigma V^T$ e, portanto, $M^T = V \Sigma U^T$, temos:

$$M^TM = (V\Sigma U^T)(U\Sigma V^T)$$

Como U é ortogonal, vale que $U^TU = I$, assim:

$$M^TM = V\Sigma(U^TU)\Sigma V^T = V\Sigma I \Sigma V^T = V\Sigma^2 V^T$$

Lembrando que na diagonalização de matrizes, uma matriz A é diagonalizável se existe uma matriz P tal que $A = PDP^{-1}$, e sabendo que V é ortogonal ($V^T = V^{-1}$), concluímos que, para a diagonalização de M^TM, as colunas de V são os autovetores de M^TM e as entradas de Σ^2 são os autovalores de M^TM, ou seja, os quadrados dos valores singulares.

Analogamente:

$$MM^T = (U\Sigma V^T)(V\Sigma U^T)$$

Como V é ortogonal, vale que $V^TV = I$, assim:

$$MM^T = (U\Sigma V^T)(V\Sigma U^T) = U\Sigma I \Sigma U^T = U\Sigma^2 U^T$$

Assim, sabendo que U é ortogonal ($U^T = U^{-1}$), concluímos que, para a diagonalização de MM^T, as colunas de U são os autovetores de MM^T e as entradas de Σ^2 são os autovalores de MM^T, ou seja, os quadrados dos valores singulares.

Também é possível ver uma clara descrição da prova original de Beltrami da existência da SVD em um caso simples na seção 2, página 5, do artigo de G. W. Stewart, "On the Early History of the Singular Value Decomposition".

Para uma melhor compreensão dessa teoria da decomposição em valores singulares de M através dos autovetores e autovalores de MM^T ou M^TM, também veja um exemplo neste tutorial do MIT.

3.1.1 Algoritmo

O algoritmo usando autovetores e autovalores, embora seja mais claro, não é bom, pois ao realizar o produto M^TM, perdemos acurácia na solução, principalmente com matrizes grandes.

Dessa forma, usaremos um algoritmo que é ainda bastante usado para a decomposição em valores singulares de uma matriz M, que é similar ao encontrado, por exemplo, no Matlab e na biblioteca de álgebra linear Lapack.

Esse algoritmo consiste em dois passos, um que simplificará a matriz M a ser decomposta e outro que fará realmente a decomposição em valores singulares:

1. O primeiro passo é reduzir a matriz M em uma matriz bidiagonal superior (todos os elementos são zero, exceto talvez os da diagonal principal e da diagonal acima dela) usando uma série de transformações de Householder. Alternativamente, aplica-se multiplicações reflexões Householder à esquerda, para zerar iterativamente as colunas, e outras à direita, para zerar iterativamente as linhas. Esse método é conhecido como Bidiagonalização de Golub-Kahan.

   Nesse passo, transformaremos a matriz M na matriz bidiagonal superior B usando duas matrizes ortogonais: U_1 (produto das multiplicações de reflexões Householder à esquerda) e V_1 (produto das multiplicações de reflexões Householder à direita).

   Veja o algoritmo de Bidiagonalização de Golub-Kahan em detalhes na seção 12.1 do artigo How to Compute the SVD do Instituto de Tecnologia de Illinois.

2. O segundo passo é diagonalizar a matriz resultante da Bidiagonalização de Golub-Kahan, B, determinando sua SVD por um processo muito parecido com o algoritmo QR. Nesse processo, conhecido como Golub–Reinsch SVD, as entradas da superdiagonal de B são zeradas usando rotações de Givens.

   O algoritmo de Decomposição em Valores Singulares de Golub–Reinsch está descrito em no capítulo 45.2 do livro "Handbook of Linear Algebra" - "Computation of the Singular Value Decomposition" da Universidade do Texas. Mais especificamente, o algoritmo é o 1-b, que faz chamadas do 1-c.

   O algoritmo 1-a é o primeiro passo aqui descrito, ou seja, a redução à forma bidiagonal usando Householder, conhecida também como algoritmo de bidiagonalização de Golub-Kahan.

Em octave, pode-se verificar a decomposição em valores singulares de uma matriz M com a seguinte chamada:

$$\text{octave}> [U, S, V] = svd(M)$$

3.2 Compressão de Imagens

Representando uma imagem de tamanho $m \times n$ por uma matriz M de tamanho $m \times n$ podemos aplicar o algoritmo de SVD em M e obter as matrizes U, V e Σ da decomposição.

Em octave, pode-se usar para ler uma imagem bitmap:

A = double(imread('lena.bmp'))-128;

Dessa forma, se a imagem A estiver no espaço de cores RGB, as diferentes cores estarão em: A(:,:,1), A(:,:,2) e A(:,:,3).

Um conceito importante é o *rank* k da matriz, que é dado pelo número de elementos não-zero da diagonal de Σ. A ideia fundamental de compressão de imagens baseada em SVD é escolher o menor número de *rank* para aproximar a matriz original.

Assim, com a imagem original sendo representada por:

$$M = U\Sigma V^T,$$

Onde:

- U é de dimensão $m \times m$;
- $\Sigma = diag(\sigma_1, \sigma_2, \ldots, \sigma_k, 0, \ldots, 0)$
- V é de dimensão $n \times n$.

Ou seja,

$$M = u_1\sigma_1 v_1^T + u_1\sigma_2 v_2^T + \cdots + u_{k-1}\sigma_{k-1}v_{k-1}^T + u_k\sigma_k v_k^T$$

,

Onde u_i e v_i são as colunas de, respectivamente, U e V.

E, podemos obter a seguinte imagem aproximada:

$$M_c = U\Sigma_c V^T,$$

Onde:

- U é de dimensão $m \times k_c$;
- $\Sigma_c = diag(\sigma_1, \sigma_2, \ldots, \sigma_{k_c})$
- V é de dimensão $k_c \times n$.

Ou seja,

$$M = u_1\sigma_1 v_1^T + u_1\sigma_2 v_2^T + \cdots + u_{k_c-1}\sigma_{k_c-1}v_{k_c-1}^T + u_{k_c}\sigma_{k_c} v_{k_c}^T$$

,

Onde u_i e v_i são as colunas de, respectivamente, U e V.

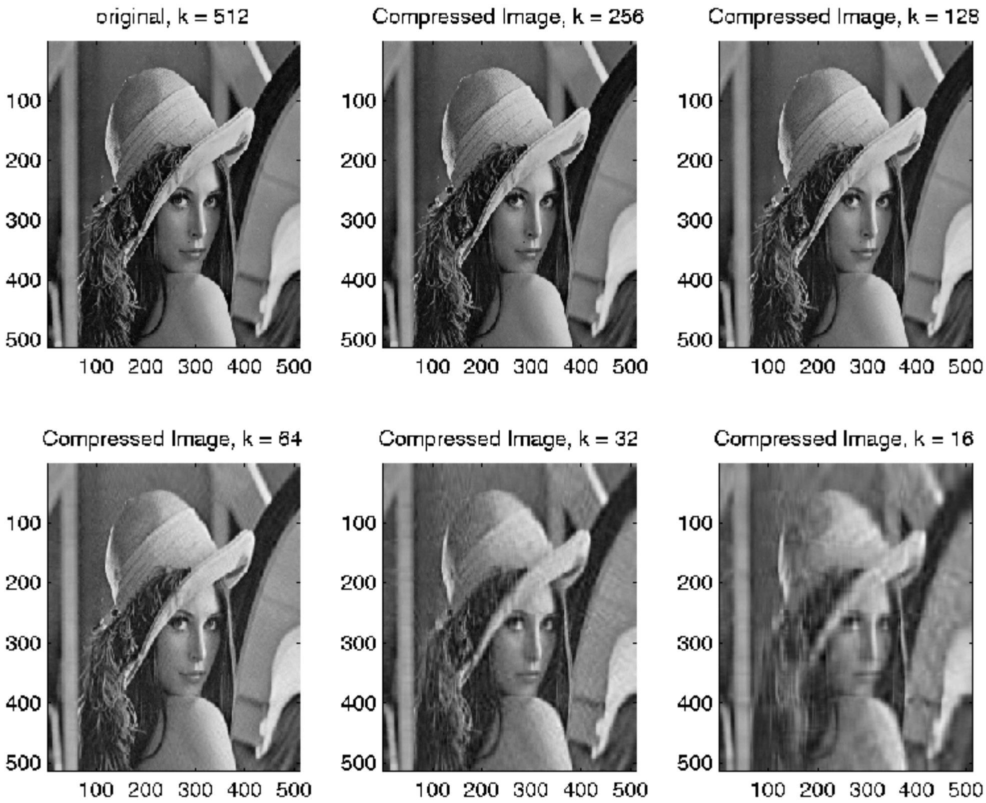

Fig. 1: *Resultados da compressão da imagem de "Lena" com diferentes valores para k_c. Fonte*

Como $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_k$, os primeiros termos da soma causam um impacto maior na soma total. Dessa forma, a compressão é, na realidade, uma aproximação da matriz original por uma soma que considera apenas os k_c primeiros termos da série. Esse é o resultado do Teorema de Eckart-Young.

Na figura 1, vemos exemplos de compressões com diferentes valores para k_c.

Leia o artigo Image Compression Using Singular Value Decomposition para um visão mais detalhada do processo.

4 Manual

Junto com o arquivo que contém seu código-fonte, submeta uma breve descrição de como compilar e usar o seu programa, inclusive com as especificações já ditas anteriormente.

5 Relatório

Elabore um relatório contendo os seguintes itens:

- Explique sucintamente o método de Decomposição em Valores Singulares;

- Descreva as principais melhorias do algoritmo implementado para SVD (Golub-Reinsch SVD) de M em relação ao método clássico, que usa os autovalores e autovetores de MM^T e M^TM.
- Faça a análise da complexidade do algoritmo de SVD implementado (Bidiagonalização de Golub-Kahan e Golub-Reinsch SVD).
- Mostre, para uma determinada imagem em RGB, diferentes compressões variando o valor do rank.

6 Fontes e Referências

- Singular Value Decomposition
- Householder Transformation
- Givens Rotation How to Compute the SVD
- Matrix Computations - Gene H. Golub,Charles F. Van Loan
- Computation of the Singular Value Decomposition
- Image Compression Using Singular Value Decomposition
- Image Compression with SVD
- Low Rank Approximation