Pedro Paulo Vezzá Campos

Compressão de imagens usando SVD

São Paulo - SP, Brasil

10 de dezembro de 2012

Pedro Paulo Vezzá Campos

Compressão de imagens usando SVD

Terceiro exercício-programa apresentado para avaliação na disciplina MAC0300, do curso de Bacharelado em Ciência da Computação, turma 45, da Universidade de São Paulo, ministrada pelo professor Walter Figueiredo Mascarenhas.

DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO
INSTITUTO DE MATEMÁTICA E ESTATÍSTICA
UNIVERSIDADE DE SÃO PAULO

São Paulo - SP, Brasil

10 de dezembro de 2012

Sumário

1 Introdução

Neste terceiro exercício-programa de MAC0300 - Métodos Numéricos da Álgebra Linear foi pedido que implementássemos um programa que fosse capaz de decompor uma matriz em seus valores singulares (Decomposição SVD) e aplicar tal algoritmo para a compressão de imagens. Neste relatório serão apresentados: Uma explicação e análise sobre SVD e seus algoritmos (Clássico e Golub-Reinsch), testes realizados e, por fim, será feita apresentada uma conclusão sobre o EP.

2 Decomposição em Valores Singulares

A Decomposição em Valores Singulares (SVD) é uma conhecida fatoração de uma matriz em três termos. Tal decomposição possui várias aplicações, variando desde processamento de sinais até estatística.

Uma SVD é definida para uma matriz M de dimensões $m \times n$ como sendo o produto

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

com as seguintes restrições:

- U é uma matriz $m \times m$. Suas colunas são conhecidas como *vetores singulares à esquerda* e são autovetores de MM^T
- Σ é uma matriz diagonal. Os valores da diagonal de Σ são conhecidos como *valores singulares de M* e são as raízes quadradas dos autovalores diferentes de zero de tanto $M^T M$ quanto MM^T.
- V^T é uma matriz $n \times n$. Suas colunas são conhecidas como *vetores singulares à direita* e são autovetores de $M^T M$

Para fins deste EP, houve a opção por implementar uma versão da SVD compacta. Nela, nem todos os autovetores são calculados, apenas os necessários para a reconstrução da matriz M a partir dos três fatores. No código implementado, a decomposição é descrita da seguinte forma:

- U é $m \times min(m,n)$ e possui colunas ortogonais
- S é $min(m,n) \times min(m,n)$ e é uma matriz diagonal contendo na diagonal principal os valores singulares
- V é $n \times min(m,n)$ e possui colunas ortogonais

2.1 Algoritmo Clássico

A decomposição em valores singulares possui como vantagem o fato que pode ser aplicado a qualquer matriz de dimensões $m \times n$ enquanto a decomposição em autovalores e autovetores só é possível para algumas matrizes quadradas. No entanto, podemos traçar paralelos entre ambas decomposições.

Dada uma SVD de uma matriz M, temos que:

$$
\begin{aligned}
M^T M &= V\Sigma^T U^T U\Sigma V^T = V(\Sigma^T\Sigma)V^T \\
MM^T &= U\Sigma V^T V\Sigma^* U^T = U(\Sigma\Sigma^T)U^T.
\end{aligned}
$$

As expressões que se encontram no lado direito das igualdades descrevem decomposições em autovalores e autovetores das expressões do lado esquerdo. Isso traz como consequência que:

- As colunas de V são autovetores de $M^T M$.
- As colunas de U são autovetores de MM^T.
- Os valores diferentes de zero de Σ são as raizes quadradas dos autovalores diferentes de zero de $M^T M$ ou MM^T.

Este algoritmo para a obtenção da decomposição em valores singulares funciona corretamente para matrizes menores e quando os valores singulares são significativamente maiores que a precisão adotada nos cálculos. Por outro lado, há uma perda de precisão inerente ao algoritmo que será explicado na 2.4, o que nos induz a buscar um algoritmo que não faça uso do cálculo de autovalores e autovetores de $M^T M$, que será apresentado na seção 2.2.

2.2 Algoritmo de Golub-Reinsch

O Algoritmo de Golub-Reinsch é um método que faz uso extensivo de rotações e reflexões para obter a SVD sem calcular $M^T M$, o que poderia trazer problemas como será apresentado em seguida. O processo é dividido em duas etapas. Na primeira fase devemos reduzir a matriz M original à forma bidiagonal utilizando reflexões de Householder aplicadas alternadamente à esquerda para zerar as colunas abaixo da diagonal principal e à direita para zerar linhas acima da superdiagonal. A segunda fase é a computação propriamente dita da SVD. Através de rotações de Givens vamos iterativamente zerando os elementos da superdiagonal enquanto acumulamos as rotações realizadas nas matrizes U e V. Este processo é repetido enquanto não for atingida uma precisão maior que uma especificada (O ε da máquina por exemplo).

2.3 Análise de Complexidade do Algoritomo de Golub-Reinsch

2.3.1 Primeira Fase: Bidiagonalização de Golub-Kahan

Como vamos aplicando refletores alternadamente à esquerda e à direita, ao final da primeira fase do cálculo da SVD são necessários n refletores à esquerda e $n-2$ refletores à direita. Podemos traçar um paralelo entre o processo de bidiagonalização e o de aplicar duas fatorações QR de Householder entrelaçadas, a primeira operando na matriz M de dimensões $m \times n$ e a outra operando na matriz M^T de dimensões $n \times m$. Assim, o custo total para a bidiagonalização é de $\sim 4mn^2 - \frac{4}{3}n^3$ flops. [1]

2.3.2 Segunda Fase: Golub-Reinsch SVD

Na segunda fase a princípio seria necessário um número infinito de rotações de Givens para que a matriz Σ convirja a uma matriz diagonal. Porém, a convergência é superlinear e em $O(nlog(|log(\varepsilon)|))$ iterações atingimos a precisão ε da máquina. Na prática, sendo ε uma constante, a convergência é dada em $O(n)$ iterações. Ainda, como a matriz é originalmente bidiagonal, são necessários apenas $O(n)$ flops por iteração, totalizando $O(n^2)$ flops para a segunda fase. Como conclusão, na prática, a primeira fase do algoritmo de Golub-Reinsch é assintoticamente mais custosa que a segunda, ditando a complexidade final do algoritmo. [1]

2.4 Melhorias do Algoritmo de Golub-Reinsch ao Método Clássico

Apesar do algoritmo clássico ser relativamente barato computacionalmente ele possui como problema o fato que valores singulares pequenos serão calculados de maneira imprecisa. Isto é uma consequência do efeito de "perda de informação através da elevação ao quadrado"que acontece quando calculamos M^TM a partir de A.

Nós podemos ter uma ideia desta perda de informação ao considerar um exemplo. Suponha que as entradas da matriz A são conhecidas com exatidão em seis casas decimais. Se A tem, digamos $\sigma_1 \approx 1$ e $\sigma_{17} \approx 10^{-3}$, então σ_{17} é razoavelmente menor que σ_1, mas ainda assim acima da precisão de $\varepsilon \approx 10^{-5}$ ou 10^{-6}. Nós gostaríamos de calcular $\sigma_1 7$ com talvez duas ou três casas de precisão. As entradas de $M^T M$ tem também precisão de aproximadamente volta de seis casas decimais. Associada com os valores singulares σ_1 e σ_{17}, $M^T M$ tem autovalores $\lambda_1 = \sigma_1^2 \approx 1$ e $\lambda_{17} = \sigma_{17}^2 \approx 10^{-6}$. Note que λ_{17} tem a mesma magnitude que os erros nas entradas de $M^T M$. Portanto não podemos esperar que λ_{17} possa ser calculado de maneira precisa. [2]

3 Testes Realizados

4 Conclusão

O trabalho ajudou os alunos a entrar em contato com a área de Computação Gráfica, pouco abordada durante a os estudos habituais de um aluno de Ciência da Computação. Com uma introdução teórica suficiente e exercícios práticos de implementação relacionados ao assunto os alunos conseguiram fixar os novos conceitos tais como convolução e diferentes métodos de filtragem de imagens. O tema gera mais interesse ainda pela sua conotação visual inerente, que torna mais palatável os resultados obtidos depois dos diversos processamentos.

Novamente, a possibilidade de utilizar uma linguagem voltada pra processamentos matemáticos, tal como Octave, a escolhida para este trabalho, simplificou problemas de implementação, permitindo aos alunos focarem nos algoritmos propriamente ditos. Ainda, a carga de teoria embutida no EP2 diminuiu com relação ao EP1, algo interessante dadas as dificuldades que os alunos enfrentaram ao iniciarem seus trabalhos no EP1.

Referências Bibliográficas

[1] TREFETHEN, L. N.; BAU, D. *Numerical Linear Algebra*. SIAM: Society for Industrial and Applied Mathematics, 1997. ISBN 0898713617. Disponível em: <http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/0898713617>.

[2] WATKINS, D. *Fundamentals of Matrix Computations*. Wiley, 2004. (Pure and Applied Mathematics: A Wiley Series of Texts, Monographs and Tracts). ISBN 9780471461678. Disponível em: <http://books.google.com.br/books?id=8t7mHZxNqIMC>.

[3] WIKIPEDIA. *Singular value decomposition — Wikipedia, The Free Encyclopedia*. 2012. [Online; accessed 10-December-2012]. Disponível em: <`http://en.wikipedia.org/w/index.php?title=Singular_value_decomposition&oldid=526553712`>.