STA 141C - Big Data & High Performance Statistical Computing **Spring 2022**

Week 6-2: Iterative methods and singular value decomposition

Lecturer: Bo Y.-C. Ning *May 05, 2022*

Last time

- PageRank problem
- Iterative method - Jacobi method

Today

- Iterative methods
- Singular value decomposition (SVD)
- Principal component analysis (PCA)

1 Iterative methods

We introduce three iterative methods: Jacobi method, Gauss-Seidel method, and Successive over-relaxation (SOR) method.

1.1 Jacobi method

Let $Ax = b$, where A be a square matrix, decompose A into a diagonal matrix D, a lower triangular matrix L and an upper triangular matrix U, such that

$$A = D + L + U.$$

The Jacobi method is then iterating

$$x^{(k+1)} = D^{-1}b - D^{-1}(L+U)x^{(k)}.$$

(Do we need to solve D to obtain D^{-1}?) One can check that this is the same as solving $x^{(k+1)}$ from

$$Dx^{(k+1)} = -(L+U)x^{(k)} + b.$$

It turns out for each x_i, it can be solved by iterating

$$x_i^{(k+1)} = \frac{b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)}}{a_{ii}}.$$

The total flops for Jacobi is about $O(n^2)$ for unstructured A for each iteration, it is faster than GE/LU if the algorithm converges in less than n iteration. For sparse matrix A, the saving is huge.

A few things to consider: How to choose the starting point for Jacobi? (Random guess; start at $1/n$ for all x_i; solution for small matrix?) When to stop the algorithm?

1.2 Gauss-Seidel method

The Gauss-Seidel method solve $x^{(k+1)}$ from

$$(D+L)x^{(k+1)} = -Ux^{(k)} + b,$$

this implies that $x^{(k+1)} = -(D+L)^{-1}Ux^{(t)} + (D+L)^{-1}b$. Thus, for each $x_i^{(k+1)}$, its solution is

$$x_i^{(k+1)} = \frac{b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)}}{a_{ii}}.$$

The Gauss-Seidel method converges for any starting point if A is symmetric and positive definite. The default convergence rate for this method is chosen to be the spectral radius of the matrix $(D+L)^{-1}U$. For Jacobi method, the default convergence rate is chosen to be the spectral radius of the matrix $D^{-1}(L+U)$. Note that comparing Jacobi with Gauss-Seidel, Jacobi is particularly attractive for parallel computing.

1.3 Successive over-relaxation (SOR)

When the Gauss-Seidel method has a slow convergence rate, a way to improve the convergence rate is the SOR method, which solve $x_i^{(k+1)}$ by solving

$$x_i^{(k+1)} = w\frac{b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)}}{a_{ii}} + (1-w)x_i^{(k)},$$

where one need to pick a value for $w \in [0,1]$ beforehand.

1.4 Conjugate gradient method

One can solve $Ax = b$ by minimizing the quadratic function $\frac{1}{2}x'A'x - b'x$. We will study this method later when we introducing the gradient descent method. For now, note that conjugate gradient and its variants are the top-notch iterative methods for solving huge, structured linear systems.

Table 1. Kershaw's results for a fusion problem.

Method	Number of iterations
Gauss Seidel	208,000
Block successive overrelaxation methods	765
Incomplete Cholesky conjugate gradients	25

2 Review of singular value decomposition (SVD)

For a rectangular matrix $A \in \mathbb{R}^{m\times n}$, let $p = \min\{m, n\}$, then we have the SVD

$$A = U\Sigma V',$$

where $U = (u_1, \ldots, u_m)$ and $V = (v_1, \ldots, v_n)$ are orthogonal matrices and $\Sigma = \text{diag}(\sigma_1, \ldots, \sigma_p)$ is a diagonal matrix such that $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_p \geq 0$. σ_is are called the *singular values*, u_is are the left singular vectors and v_is are the right singular vectors.

The matrix Σ is not a square matrix, one can define thin SVD, which factorizes A as

$$A = U_n\Sigma_n V' = \sum_{i=1}^{n} \sigma_i u_i v_i',$$

where $U_n \in \mathbb{R}^{m\times n}$, $U_n'U_n = I_n$, $\Sigma = \text{diag}(\sigma_1, \ldots, \sigma_n)$. This is for $m > n$, if $m < n$, then we let $V \in \mathbb{R}^{m\times n}$,

The following properties are useful: for $\sigma(A) = (\sigma_1, \ldots, \sigma_p)'$, the rank of A is the number of nonzero singular values denoted as $\|\sigma(A)\|_0$. The Frobenius norm of A, $\|A\|_F = (\sum_{i=1}^{p} \sigma_i^2)^{1/2} = \|\sigma(A)\|_2$, and the spectrum norm of A, $\|A\|_2 = \sigma_1 = \|\sigma(A)\|_\infty$. Using the fact that U, V are both orthogonal matrices

$$A'A = V\Sigma U'U\Sigma V' = V\Sigma^2 V',$$
$$AA' = U\Sigma V'V\Sigma U' = U\Sigma^2 U'$$

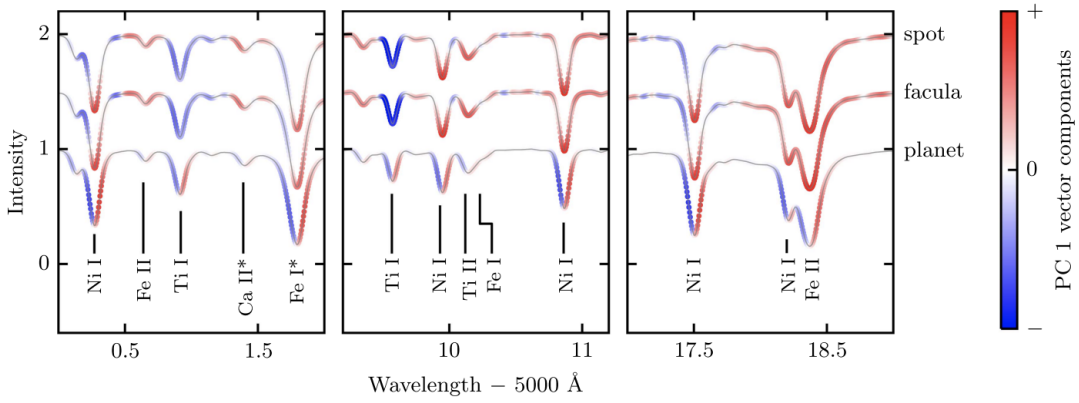
Last, the eigen-decomposition for a real symmetric matrix is $B = W\Lambda W'$, where $\Lambda = \text{diag}(\lambda_1, \ldots, \lambda_n)$, which is the SVD of B.

3 Applications for SVD

1. Principal component analysis and dimension deduction.

Let $X \in \mathbb{R}^{n\times p}$ be a centered data matrix, perform SVD on $X = U\Sigma V'$. The linear combinations $\tilde{x}_i = Xv_i$ are the principal components (PCs) with variance σ_i^2.

Dimension deduction: reduce dimensionality p to $q \leq p$ and use the first few PCs $\tilde{x}_i, \ldots, \tilde{x}_q$ in downstream analysis. Used in medical studies, astronomy, etc.



Davis et al. (2017). *Insights on the Spectral Signatures of Stellar Activity and Planets from PCA*. The Astrophysical Journal.

2. Low rank approximation in image/data compression.

Goal: find Y, $\min_{\text{rank}(X)=r}\|X-Y\|_F^2$. By Eckart-Young theorem, $Y=\sum_{i=1}^r \sigma_i u_i' v_i$ with optimal value $\sum_{i=1}^r \sigma_i^2$, where (σ_i, u_i, v_i) are singular values and vectors of X.

Gene Golub's 2691×598 picture requires $2691\times 598\times 6 = 9,655,308$ bytes (RGB 16 bit per channel). Rank 120 approximation requires $120\times(2691+598)\times 6 = 2,368,080$ bytes. Rank 50 approximation requires $50\times(2691+598)\times 6 = 986,700$ bytes. Rank 12 approximation requires $12\times(2691+598)\times 8 = 236,808$ bytes.

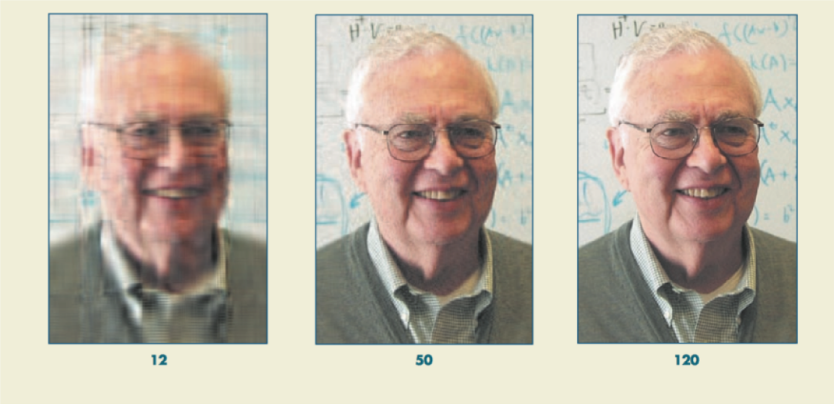


Figure 2. Rank 12, 50, and 120 approximations to a rank 598 color photo of Gene Golub.

4 Method for computing SVD: Power method

To start, let's assume $A\in\mathbb{R}^{n\times n}$ is a symmetric and p.s.d. matrix, the power method for obtaining the largest eigenvalue is given as:

1) Choose an initial guess of $q^{(0)}$ (non-zero);

2) Repeat $k=1,\ldots,K$,

$$z^{(k)} = Aq^{(k-1)}$$
$$q^{(k)} = \frac{z^{(k)}}{\|z^{(k)}\|_2};$$

3) Output: $\lambda_1 \leftarrow {q^{(K)}}' A q^{(K)}$.