Lecture 1: Introduction to Data Analytics and Linear Algebra in Big Data Analytics

Haesun Park

School of Computational Science and Engineering
Georgia Institute of Technology
Atlanta GA, U.S.A.

SIAM Gene Golub Summer School, Aussois France, June 18, 2019

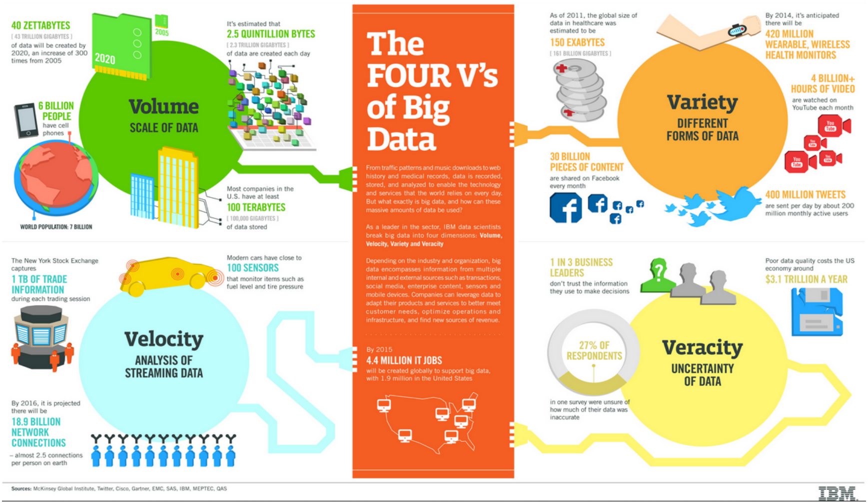
This work was supported in part by NSF, Homeland Security, DARPA

Overview of Lectures, June 18-19

- Introduction to Data Analytics. Linear Algebra in Data Analytics (focus: Low Rank Approximation)
- Constrained Low Rank Approximation (CLRA) and Data Analytic Tasks (focus: Dimension Reduction and Clustering)
- Constrained Low Rank Approximation:
  - Nonnegative Matrix Factorization (NMF) for dimension reduction, clustering, and topic modeling
  - Symmetric NMF for graph clustering and community detection
  - JointNMF for clustering utilizing content/attributes and connection information
- Applications in text and social network analyses

Overview of Lecture 1

- Introduction to Data Analytics : Challenges
- Role of Linear Algebra in Data Analytics, specifically, Low Rank Approximation (LRA) : SVD and Rank



- Volume: Large number of data items, High-dimensional, Complex relationships
- Variety: Of heterogeneous formats, sources, reliability
- Velocity: Time varying, dynamic,...
- Veracity: Noisy, varying quality, errors and missing values are inevitable in real data set
  ... Vast majority of data is unstructured: Text

(Reproduction from a slide courtesy of IBM)

The Challenge

- Transform the data into knowledge (understanding, insight), making it useful to people
- Ways to get to Knowledge: Automated algorithms and Visualization
  - Faster methods to solutions
  - Accurate solutions/less errors
  - Better understanding/interpretation

DATA

- Data is taken from some phenomena from the world
- Data refers to qualitative or quantitative attributes of a variable or set of variables
- Data is the lowest level of abstraction from which information and then knowledge are derived
- Examples: Text data from news articles, image data from satellites, video data from surveillance cameras, connection data from social network
- How to provide data to vector-space based algorithms:
    - Data Representation in matrices or tensors
    - Feature (attribute)-data relationship or data-data relationship
    - *Dimension*: often refers to the number of features/attributes

Major Tasks in Data Analytics

- Dimension Reduction
- Clustering
- Classification
- Regression
- Trend analysis
- ...

Numerical Linear Algebra

- Problems:
  - Linear systems
  - Least Squares
  - Eigenvalue problems
- Methods:
  - Direct: often involves Decomposition (Factorization) of a matrix, to transform the given problem into another problem which is easeir to solve: LU, QR, SVD, EVD, ...
  - Iterative
- Since the main topic is Constrained Low Rank Approximation, we will first focus on Rank and SVD

Singular Value Decomposition

For any matrix $A \in R^{m \times n}$, there exist matrices U, V, Σ such that
$A = U\Sigma V^T$,
where $U \in R^{m \times m}, U^T U = I_m$, $V \in R^{n \times n}, V^T V = I_n$,
$\Sigma = diag(\sigma_1, \sigma_2, \cdots) \in R^{m \times n}$ where

$$\Sigma = \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_n \\ & 0 & \end{bmatrix} \text{ when } m \geq n,$$

$$\Sigma = \begin{bmatrix} \sigma_1 & & & \\ & \ddots & & 0 \\ & & \sigma_n & \end{bmatrix} \text{ when } m \leq n,$$

and $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq 0$ are singular values .

Properties of SVD

Suppose for $A \in R^{m\times n}(m \geq n)$, we have its SVD $A = U\Sigma V^T$.

- $A^T A = V\Sigma^T U^T U\Sigma V^T = V\Sigma^T\Sigma V^T$ where $\Sigma^T\Sigma = diag(\sigma_1^2, \cdots, \sigma_n^2)$
- $AA^T = U\Sigma V^T V\Sigma^T U^T = U\Sigma\Sigma^T U^T$ where $\Sigma^T\Sigma = diag(\sigma_1^2, \cdots, \sigma_n^2, 0, \cdots, 0)$
- If $\Sigma = diag(\sigma_1, \cdots, \sigma_n) = diag(\sigma_1, \cdots, \sigma_r, 0, \cdots, 0)$, i.e. $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > \sigma_{r+1} = \cdots = \sigma_n = 0$, then $rank(A) = r$
- With $A = U\Sigma V^T = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \end{bmatrix}$ where

  $\Sigma_1 = diag(\sigma_1, \cdots, \sigma_r)$ with $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$,

  $Range(A) = span(U_1)$, $Null(A) = span(V_2)$.

  $Range(A^T) = span(V_1)$, $Null(A^T) = span(U_2)$.

QR Decomposition

For any matrix $A \in R^{m \times n}$**,** $\exists$ **orthogonal matrix** $Q \in R^{m \times m}$ $(Q^T Q = I_m)$ **and upper triangular matrix** $R \in R^{n \times n}$**, s.t.** $A = Q \begin{bmatrix} R \\ 0 \end{bmatrix}$**.**

Theorem. If $A = [a_1 \cdots a_n] \in R^{m \times n}$ has $rank(A) = n$, and $A = Q \begin{pmatrix} R \\ 0 \end{pmatrix}$, where $Q = (\underbrace{Q_1}_{n} \ \underbrace{Q_2}_{m-n}) = [q_1 \cdots q_n]$,

then $A = Q_1 R$ and

- $span\{a_1 \cdots a_k\} = span\{q_1 \cdots q_k\}$, for all $k = 1, \cdots, n$.
- $Range(A) = Range(Q_1)$ and $Range^{\perp}(A) = Range(Q_2)$.
- R^T in the QRD of A is the Cholesky factor of $A^T A$

SVD and QRD

Main differences between SVD and QRD?

Least Squares Problem - QRD

Linear System $Ax = b$ where $A : n \times n$ nonsingular and $b : n \times 1$
Least Squares $Ax \approx b$ where $A : m \times n$ with $m \geq n$ and $b : m \times 1$
For solving least squares (LS) problem, we need **orthogonalization** to reduce matrices to canonical forms : **QR factorization (decomposition)** or **SVD**.
$\|Ax - b\|_2 = \|Q^T Ax - Q^T b\|_2$ for any orthogonal matrix Q $(Q^T Q = I)$

Suppose there is an orthogonal matrix Q, $Q^T A = \begin{pmatrix} R \\ 0 \end{pmatrix}$, then

$\|Q^T Ax - Q^T b\|_2 = \| \begin{pmatrix} R \\ 0 \end{pmatrix} x - \begin{pmatrix} c \\ d \end{pmatrix} \|_2 = \| \begin{pmatrix} Rx - c \\ -d \end{pmatrix} \|_2$ where

$Q^T b = \begin{pmatrix} c \\ d \end{pmatrix}$
Solution x is obtained by solving $Rx = c$ when $rank(A) = n$
(When $rank(A) = n$, $rank(R) = n$)

Least Squares Problem – SVD

Solving LS
$\min_x \|Ax - b\|_2$, $A \in R^{m\times n}$, $b \in R^{m\times 1}$, $m \geq n$.

Let the SVD of A be

$$A = U\Sigma V^T = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_1 & V_2 \end{bmatrix}^T$$

where $U_1 \in R^{m\times r}$, $U_2 \in R^{m\times(m-r)}$, $\Sigma_1 = diag(\sigma_1, \cdots, \sigma_r)$,
$\sigma_1 \geq \cdots \geq \sigma_r > 0$,
and $V_1 \in R^{n\times r}$, $V_2 \in R^{n\times(n-r)}$,i.e. $rank(A) = r$.

$$
\begin{aligned}
\|Ax-b\|_2 &= \left\|U\Sigma V^T x - b\right\|_2 = \left\|U^T(U\Sigma V^T x - b)\right\|_2 \\
&= \left\|\Sigma V^T x - U^T b\right\|_2 \\
& \begin{cases} \text{Letting } \; V^T x = \begin{pmatrix} V_1^T \\ V_2^T \end{pmatrix} x = \begin{pmatrix} y \\ z \end{pmatrix} \begin{matrix} \} & r \\ \} & n-r \end{matrix} \\ U^T b = \begin{pmatrix} U_1^T \\ U_2^T \end{pmatrix} b = \begin{pmatrix} c \\ d \end{pmatrix} \begin{matrix} \} & r \\ \} & m-r \end{matrix} \end{cases} \\
&= \left\| \begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} y \\ z \end{bmatrix} - \begin{bmatrix} c \\ d \end{bmatrix} \right\| = \left\| \begin{bmatrix} \Sigma_1 y - c \\ -d \end{bmatrix} \right\|_2
\end{aligned}
$$

Least Squares Problem – SVD

Since $\Sigma_1 \in R^{r \times r}$, non-singular, we can find the unique solution for $\Sigma_1 y - c = 0 \iff \Sigma_1 y = c \iff y = \Sigma_1^{-1} c$.

Letting $r(x) = \|Ax - b\|_2$, the residual $r(x_{LS}) = \|d\|$ where x_{LS} is the LS solution.

The solution is

$$x_{LS} = V \begin{bmatrix} y \\ z \end{bmatrix}$$

where $y = \Sigma_1^{-1} c$ and z can be anything. (if $rank(A) = n$, z is null).

$$x_{LS} = \begin{pmatrix} V_1 & V_2 \end{pmatrix} \begin{pmatrix} y \\ z \end{pmatrix} = V_1 y + V_2 z$$

Note $V_2 z \in null(A)$.

When $z = 0$, $x_{LS} = V \begin{bmatrix} \Sigma_1^{-1} c \\ 0 \end{bmatrix}$ is called **minimum-norm solution**.

Least Squares Problem – SVD

In some applications, just need to compute L_2 norm of the residual vector.

$\Longrightarrow$ Can be done **WITHOUT** computing the solution vector x.

$r = Ax_{LS} - b$
$\|r\|_2 = \|Ax_{LS} - b\|_2$

$$\begin{aligned} r &= U\Sigma V^T x_{LS} - b \\ &= U \begin{pmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} y \\ z \end{pmatrix} - U \begin{pmatrix} c \\ d \end{pmatrix} \\ &= U \begin{pmatrix} \Sigma_1 y - c \\ -d \end{pmatrix} \\ &= U \begin{pmatrix} 0 \\ -d \end{pmatrix} \end{aligned}$$

$\therefore \|r\|_2 = \|U \begin{pmatrix} 0 \\ -d \end{pmatrix}\|_2 = \|d\|_2.$

Rank Decision in LS and SVD

$$A = U\Sigma V^T = U \begin{bmatrix} 1 & & & \\ & 0.5 & & \\ & & 10^{-14} & \\ & & & 10^{-16} \\ & & & \\ & & & \end{bmatrix} V^T$$

Depending on rank decision (2 or 3 or 4?), we obtain very different solutions.

If we consider the tolerance ϵ s.t. $10^{-16} < \epsilon$, and $rank(A) = 3$,

$$\Sigma_1 = \begin{bmatrix} 1 & & \\ & 0.5 & \\ & & 10^{-14} \end{bmatrix}, y = \begin{bmatrix} 1 & & \\ & 2 & \\ & & 10^{14} \end{bmatrix} \begin{bmatrix} x \\ x \\ x \end{bmatrix}.$$

If we consider the tolerance ϵ s.t. $10^{-14} < \epsilon$, and $rank(A) = 2$,

$$\Sigma_1 = \begin{bmatrix} 1 & \\ & 0.5 \end{bmatrix}, y = \begin{bmatrix} 1 & \\ & 2 \end{bmatrix} \begin{bmatrix} x \\ x \end{bmatrix}.$$

Rank Decision in LS and SVD

Min-norm solution:

the first case, $x_{LS} = V \begin{bmatrix} \begin{bmatrix} 1 & & \\ & 2 & \\ & & 10^{14} \end{bmatrix} \begin{bmatrix} x \\ x \\ x \end{bmatrix} \\ 0 \end{bmatrix}$

the second case, $x_{LS} = V \begin{bmatrix} \begin{bmatrix} 1 & \\ & 2 \end{bmatrix} \begin{bmatrix} x \\ x \end{bmatrix} \\ 0 \end{bmatrix}$

Determination of numerical rank can be difficult. Usually we find a large gap. $\begin{bmatrix} 1 & & & & \\ & 10^{-1} & & & \\ & & 10^{-2} & & \\ & & & 10^{-3} & \\ & & & & \ddots \end{bmatrix}$ no gap?

Ex.

$$A = \begin{bmatrix} 1 & -1 & -1 & -1 \\ & 1 & -1 & -1 \\ & & 1 & -1 \\ & & & 1 \end{bmatrix}, \ \det(A) = 1$$

$$A^{-1} = \begin{bmatrix} 1 & 2 & 2 & 4 \\ & 1 & 2 & 2 \\ & & 1 & 2 \\ & & & 1 \end{bmatrix}$$

In general $A^{-1}(1, n) = 2^{n-2}$ when $A = \begin{bmatrix} 1 & -1 & -1 & -1 \\ & 1 & -1 & -1 \\ & & 1 & -1 \\ & & & 1 \end{bmatrix}$

$K_1(A) = \|A\|_1 \left\|A^{-1}\right\|_1 \approx n \cdot 2^{n-2}$

The reason of "bad" solution:

1. Algorithm is bad? (unstable)
2. Problem difficult? (ill-conditioned)

Reduced Rank in Data Analytics

- In data analytics, reduced rank k of interest is the reduced dimension or the number of clusters, topics, communities
- Often k is much smaller than the rank of the data matrix r
- However, an optimal reduced rank k in data analytics is not easy to determine either: the optimal number of clusters? the optimal reduced dimension ?
- Will assume k is given and $k << r$: often requres very severe low rank approximation

QRD with Column Pivoting can Reveal Rank

If $A = QR$ **and** $rank(A) = n$**, then**

$span\{a_1, \cdots, a_k\} = span\{q_1, \cdots, q_k\}$, $1 \leq k \leq n$ **where**

$A = \begin{bmatrix} a_1 & \cdots & a_n \end{bmatrix}$, $Q = \begin{bmatrix} q_1 & \cdots & q_n \end{bmatrix}$.

Why QRD with Column Pivoting?

E.g. $A = \begin{bmatrix} 1 & 1 & 1 \\ & & 1 \\ & & 1 \end{bmatrix}$. $rank(A) = 2$.

Consider QRD of A

$$A = QR = \begin{bmatrix} 1 & & \\ & 1 & \\ & & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ & & 1 \\ & & 1 \end{bmatrix}$$

Although $rank(A) = 2$, we don't have $Range(A) = span\{q_i, q_j : i \neq j\}$. **QRD with C.P. can help us to maintain** $span\{a_1, \cdots, a_k\} = span\{q_1, \cdots, q_k\}$ **in rank deficient case.**

For any $A \in R^{m \times n}$, **QRD with Column Pivoting** computes

$$A\Pi = Q \begin{pmatrix} R \\ 0 \end{pmatrix}$$

, where

$$R = \begin{pmatrix} R_{11} & R_{12} \\ \underbrace{0}_{(n-r)\times r} & \underbrace{0}_{(n-r)\times(n-r)} \end{pmatrix}$$

where $R_{11} \in \mathbb{R}^{r \times r}$ is upper triangular, $R_{12} \in \mathbb{R}^{r \times n-r}$,
$r = rank(A) = rank(R) = rank(R_{11})$.

$Q \in R^{m \times m}$, orthogonal; $\Pi \in R^{n \times n}$, permutation.

$$A\Pi = Q\begin{bmatrix} R \\ 0 \end{bmatrix} = Q\begin{bmatrix} R_{11} & R_{12} \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \Longleftrightarrow A = Q\begin{bmatrix} R_{11} & R_{12} \\ 0 & 0 \\ 0 & 0 \end{bmatrix}\Pi^T.$$

$$\|Ax - b\|_2 = \left\| Q\begin{bmatrix} R_{11} & R_{12} \\ & \end{bmatrix}\Pi^T x - b\right\|_2 = \left\|\begin{bmatrix} R_{11} & R_{12} \\ & \end{bmatrix}\Pi^T x - Q^T b\right\|_2$$

Letting $\Pi^T x = \begin{bmatrix} y \\ z \end{bmatrix} \begin{matrix} \} & r \\ \} & n-r \end{matrix}$, $Q^T b = \begin{bmatrix} c \\ d \end{bmatrix} \begin{matrix} \} & r \\ \} & m-r \end{matrix}$,

$$\|Ax - b\|_2 = \left\|\begin{bmatrix} R_{11} & R_{12} \\ 0 & 0 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} y \\ z \end{bmatrix} - \begin{bmatrix} c \\ d \end{bmatrix}\right\|_2 = \left\|\begin{pmatrix} R_{11}y + R_{12}z - c \\ -d \end{pmatrix}\right\|_2$$

Least Squares Problem – Rank Deficient

z can be anything, and y can be chosen so that $R_{11}y = c - R_{12}z$.
Can set $z = 0$, then y satisfies $R_{11}y = c$.

$x = \Pi \begin{bmatrix} R_{11}^{-1}(c - R_{12}z) \\ z \end{bmatrix}$ where z is free.

If we set $z = 0$, we get basic solution $x = \Pi \begin{bmatrix} R_{11}^{-1}c \\ 0 \end{bmatrix}$.

When $rank(A) = n$, $A\Pi = Q \begin{bmatrix} R_{11} \\ 0 \end{bmatrix}$ and $R_{11} = R$.

QRD with Column Pivoting

A **can be nearly rank deficient without any** $fl(R_{22}^{(k)})$ **being very small.**

From V. Kahan

$$T_n(c) = diag(1, s, s^2, \cdots s^{n-1}) \begin{bmatrix} 1 & -c & \cdots & -c \\ & 1 & \ddots & \vdots \\ & & \ddots & -c \\ & & & 1 \end{bmatrix}$$

Least Squares Problem – Rank Deficient

$$T_5(c) = \begin{bmatrix} 1 & -c & -c & -c & -c \\ & s & -cs & -cs & -cs \\ & & s^2 & -cs^2 & -cs^2 \\ & & & s^3 & -cs^3 \\ & & & & s^4 \end{bmatrix}$$

$k=1$: all columns have norm 1 $\longrightarrow$ no permutation, no annihilation.

$k=2$: $c^2s^2 + s^4 = s^2(c^2+s^2) = s^2$ all columns have same norm $\longrightarrow$ no permutation, no annihilation.

For any k, $\left\| R_{22}^{(k+1)} \right\|_F \geq s^{n-1}$.

$T_{100}(0.2)$ has no very small trailing principal submatrix since $\left\| R_{22}^{(k+1)} \right\|_F \geq s^{99} \approx 0.13$, but $\sigma_{100} \approx 10^{-8}$.

QRD with column pivoting is not completely reliable for detecting near rank deficiency.

However in practice, QRD with column pivoting works well.

Eigenvalue Problem

Given a matrix $A \in C^{n\times n}$, $\exists$ n scalars λ_i(eigenvalues), n vectors $v_i \neq 0$(eigenvectors), s.t. $Av_i = \lambda_i v_i$.

Set of eigen values of A: $\lambda\{A\}$
Eigenvalues are the roots of **characteristic polynomial**
$P_A(\lambda) = \det(\lambda I - A)$ or $\det(A - \lambda I)$
e.g. $A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$, $\lambda I - A = \begin{bmatrix} \lambda - 1 & -2 \\ -3 & \lambda - 4 \end{bmatrix}$,
$\det(\lambda I - A) = (\lambda - 1)(\lambda - 4) - 6 = 0$

Eigenvalues of a diagonal matrix are the diagonal elements.
Eigenvalues of a triangular matrix are the diagonal elements.
$p(\lambda) = \det(\lambda I - A)$ has n roots.

Eigenvalue Problem

Definition: Two matrices A and B are **similar**, if $B = X^{-1}AX$ for a nonsingular matrix X. X is called **similarity transformation**.

Theorem. If A and B are similar, i.e. $\exists X$ is nonsingular, s.t. $B = X^{-1}AX$, then $\lambda\{A\} = \lambda\{B\}$.

Note that here X is not necessarily unitary.

Proof: $P_A(\lambda) = \det(\lambda I - A)$,
$P_B(\lambda) = \det(\lambda I - X^{-1}AX) = \det(X^{-1}(\lambda I - A)X) =$
$\det(X^{-1})\det(\lambda I - A)\det(X) = \det(\lambda I - A)$ as
$1 = \det(I) = \det(X^{-1}X) = \det(X^{-1})\det(X)$

To preserve eigen values, we have to use similarity transformations.

Schur Decomposition: For any $B \in C^{n \times n}$, there exists a unitary matrix $Q \in C^{n \times n}$ s.t. $Q^H BQ = T$ where $T \in C^{n \times n}$ is upper triangular and the diagonal elements of T are the eigenvalues of B.
Symmetric EVD: For any $B \in R^{n \times n}$ with $B^T = B$, there exists an orthogonal matrix $Q = \begin{bmatrix} q_1 & \cdots & q_n \end{bmatrix}$ s.t. $Q^T BQ = \Lambda = diag(\lambda_1, \cdots, \lambda_n)$, where λ_i are eigenvalues and q_i are eigenvectors.

$$Q^T BQ = \Lambda \Longleftrightarrow BQ = Q\Lambda \Longleftrightarrow$$

$$B \begin{bmatrix} q_1 & \cdots & q_n \end{bmatrix} = \begin{bmatrix} q_1 & \cdots & q_n \end{bmatrix} \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{bmatrix}$$

For a matrix A, $\|A\|_2 = \sigma_1(A)$ where $A = U\Sigma V^T \in R^{m\times n}$, $U^T U = V^T V = I$, $\Sigma = diag(\sigma_1, \cdots, \sigma_n)$, $\sigma_1 \geq \cdots \geq \sigma_n > 0$

$A^T A = V\Sigma^T \Sigma V^T$. Largest eigenvalue of $A^T A = \sigma_1^2$, $\|A\|_2 = \sqrt{(\text{largest eigen value of } A^T A)}$

Consider $Ax = b$, $A \in R^{n\times n}$. $Cond_p(A) = \|A\|_p \left\|A^{-1}\right\|_p$, $Cond_2(A) = \|A\|_2 \left\|A^{-1}\right\|_2 = \sigma_1 \times ?$

Assume $rank(A) = n$, $A = U\Sigma V^T$, $A^{-1} = V\Sigma^{-1}U^T$, hence $\left\|A^{-1}\right\|_2 = \frac{1}{\sigma_n}$,

$$Cond_2(A) = \sigma_1/\sigma_n$$

.

Pseudo-Inverse from SVD

Assume $A \in R^{m \times n}$has its SVD

$$A = U\Sigma V^T = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \\ 0 & 0 \\ & \end{bmatrix} \begin{bmatrix} V_1 & V_2 \end{bmatrix}^T \text{ where}$$

$\Sigma_1 = diag(\sigma_1, \cdots, \sigma_r)$, $\sigma_1 \geq \cdots \geq \sigma_r > 0$

The pseudo-inverse is $A^+ = V \begin{bmatrix} \Sigma_1^{-1} & \\ & \end{bmatrix} U^T$

which is the unique minimal Frobenius norm solution to $\min_{X \in \mathbf{R}^{n \times m}} \|AX - I_m\|_F$.

If $rank(A) = n$, $A^+ = (A^T A)^{-1} A^T$

If $rank(A) = n = m$, $A^+ = A^{-1}$

Moore-Penrose pseudo-inverse A^+ for $A \in \mathbf{R}^{m \times n}$ is a unique matrix X that satisfies:

1. $AXA = A$ 2. $XAX = X$ 3. $(AX)^T = AX$ 4. $(XA)^T = XA$

Note: LS solution for $\min \|Ax - b\|_2$: $x_{LS} = A^+ b$

For any matrix $A \in R^{m \times n}$, $\exists$ matrices U, V, Σ such that

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

where $U \in R^{m \times m}, U^T U = I_m$; $V \in R^{n \times n}, V^T V = I_n$, $\Sigma \in R^{m \times n}$ such that

$$\Sigma = \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_n \\ & 0 & \end{bmatrix} \text{ when } m \geq n, \; \Sigma = \begin{bmatrix} \sigma_1 & & & \\ & \ddots & & 0 \\ & & \sigma_m & \end{bmatrix}$$

when $m \leq n$

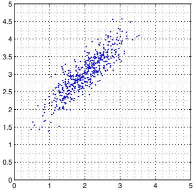
Let $U \Sigma V^T = \begin{bmatrix} U_k & \hat{U}_k \end{bmatrix} \begin{bmatrix} \Sigma_k & 0 \\ 0 & \hat{\Sigma}_k \end{bmatrix} \begin{bmatrix} V_k & \hat{V}_k \end{bmatrix}^T$ and

$A_k = U_k \Sigma_k V_k^T$: Truncated SVD

Then $\min_{rank(B)=k} \|A - B\|_F = \|A - A_k\|_F$ for $k \leq rank(A)$

Image Compression, Text Analysis (LSI), Signal Processing, ...

Consider data points in a two-dimensional space:



How can we use one variable to describe these data points?

Principal Component Analysis

Input: Data matrix $A_{m\times n}$ (m features, n data items)

Method 1 to compute PCA

1. Center the data matrix, and obtain $\tilde{A} = A - \frac{1}{n}Aee^T$ where $e = ones(n, 1)$
2. Compute SVD: $\tilde{A} = U\Sigma V^T$
3. Use U^T to transform centered data: $\tilde{A} \rightarrow U^T\tilde{A}$

Method 2 to compute PCA

1. Compute covariance matrix Ω from centered data: $\Omega = \tilde{A}\tilde{A}^T$
2. Compute SymEVD of $\Omega = U\Lambda U^T$
3. Use U^T to transform centered data: $\tilde{A} \rightarrow U^T\tilde{A}$

Dimension reduction by SVD computes SVD of A, not $\tilde{A}$

Avoid Squaring Matrices if possible!

- Example: $A = \begin{bmatrix} 1 & 1 \\ 10^{-3} & \\ & 10^{-3} \end{bmatrix}$, $b = \begin{bmatrix} 2 \\ 10^{-3} \\ 10^{-3} \end{bmatrix}$

  $x_{LS} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$, $K_2(A) \approx 1.4 \times 10^3$.
  Assume $\beta = 10,\ t = 6$, chopped arithmetic.

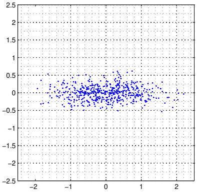
  $fl(A^T A) = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$, $rank(A) = 2,\ rank(fl(A^T A)) = 1$.

  Assume $\beta = 10,\ t = 7,\ fl(A^T A) = \begin{bmatrix} 1 + 10^{-6} & 1 \\ 1 & 1 + 10^{-6} \end{bmatrix}$,
  $\hat{x} = \begin{bmatrix} 2.00001 \\ 0 \end{bmatrix}$ where $\hat{x}$ is solution for $fl(A^T A)x = fl(A^T b)$.

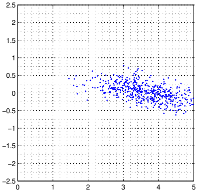
  $\frac{\|\hat{x} - x_{LS}\|_2}{\|x_{LS}\|_2} \approx \mu K_2(A^T A) = \mu (1.4 \times 10^3)^2$.

The previous example on two-dimensional data:

After PCA:



After SVD directly applied to A (instead of $\bar{A}$):

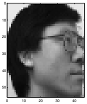


PCA and SVD for Image Compression

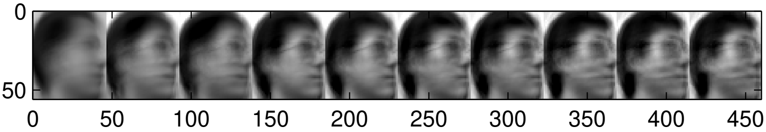
In a face data set, we have $n = 575$ images, each with $m = 56 \times 46 = 2576$ pixels.

We want to find lower rank approximation of the data matrix $A_{2576 \times 575}$ with $k = 2, 4, \cdots, 20$.

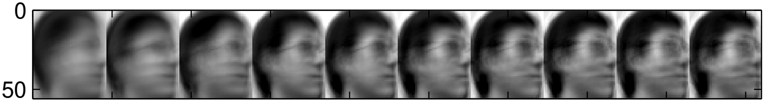
One of the original images:



After PCA (rank-k approximation of the covariance matrix):

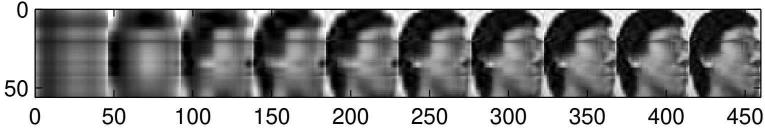


After SVD (rank-k approximation of the data matrix):



SVD for Image Compression (of *one* image)

Use a matrix $A_{56\times46}$ to represent one image.
Again, we use SVD to find the best rank-k approximation of A.
The images corresponding to best rank-k approximations $(k = 1, 2, \cdots, 10)$:

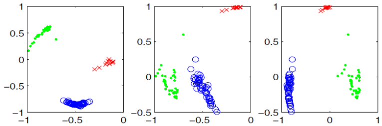


Apply SVD to the term-document matrix.
An example of term-document matrix: (from Wikipedia)

1. D1: "I like databases"
2. D2: "I hate hate databases"
3. ...

	D1	D2	...
I	1	1	...
like	1	0	...
hate	0	2	...
databases	1	1	...
⋮	⋮	⋮	⋱

LSI extracts k latent semantics represented by k orthogonal basis vectors: [Xu et al, 2003]



(b) Data distribution in the SVD subspace of $E_1 - E_2$, $E_1 - E_3$, and $E_2 - E_3$, respectively.

where E_1, E_2, E_3 are the first 3 columns of U in the SVD of term-document matrix A.

Orthogonal Procrustes Problem

$$\min_{Q, Q^T Q = I} \|AQ - B\|_F, \; A, B \in R^{m \times n}.$$

Solution is obtained from the Polar Decomposition of $A^T B$.

Polar Decomposition of a matrix $A \in \mathbf{R}^{m \times n}$ is:
$A = UP$
where $U \in \mathbf{R}^{m \times n}$ has orthonormal columns and
$P \in \mathbf{R}^{n \times n}$ is symmetric positive semidefinite.

Polar Decomposition can be computed from SVD:
$A = U \Sigma V^T = (UV^T)(V \Sigma V^T)$

QR Algorithm for Symmetric EVD

- Reduce A ($A = A^T$) to a tridiagonal matrix T: $U^T AU = T$, where U is an orthogonal matrix.
- Repeat:
  - Choose λ as an approximate eigenvalue of T
  - Compute QRD of $T - \lambda I$: $T - \lambda I = QR$,
  - $T_{new} := RQ + \lambda I$
- T_{new} is similar to T
- QRD of T is very fast: apply Givens rotations to make sub-diagonal entries of T zero
- Shift possibilities: $\lambda = T_{nn}$ or $\lambda = \mu$ where μ is the eigenvalue of $T(n-1:n, n-1:n)$ that is closer to T_{nn} (Wilkinson shift).
- Complexity of QR algorithm for Sym. EVD: $O(n^2)$ without eigenvectors and $O(n^3)$ with eigenvectors.

Jacobi Algorithm for Symmetric EVD

$A \in R^{n\times n}$, $A^T = A$, $Q^T AQ = D = diag(\lambda_1, \cdots, \lambda_n)$

1. QR algorithm, faster
2. Jacobi algorithm, **easy to parallelize**

After each step, the matrix becomes "more diagonal".

$A = \begin{bmatrix} x & y \\ y & z \end{bmatrix} \in R^{2\times 2}, \begin{bmatrix} c & -s \\ s & c \end{bmatrix} \begin{bmatrix} x & y \\ y & z \end{bmatrix} \begin{bmatrix} c & s \\ -s & c \end{bmatrix} =$
$\begin{bmatrix} ? & 0 \\ 0 & ? \end{bmatrix}$
$\Rightarrow y(c^2 - s^2) + (x - z)cs = 0.$

Jacobi Algorithm

A measure to check how close a matrix is to a diagonal form:

$$off(A) = \sqrt{\sum_{j=1}^{n} \sum_{i=1, i \neq j}^{n} a_{ij}^2} = \|A\|_F^2 - \sum_{i=1}^{n} a_{ii}^2$$

- Jacobi algorithm decreases $off(A)$?

Let $B = J^T AJ$, where $J = J(p, q, \theta)$. $Off^2(B) = \|B\|_F^2 - \sum_{i=1}^{n} b_{ii}^2 = \|A\|_F^2 - \left(\sum_{i=1}^{n} a_{ii}^2 + 2a_{pq}^2\right) = off^2(A) - 2a_{pq}^2$, where (p, q) is two entries zeroed out.
Given that $a_{pq} \neq 0$, we have $off^2(B) \leq off^2(A)$ after 1 step.

Algorithms for SVD: Discussion

How do you use the ideas of QR algorithm or Jacobi algorithm for SymEVD to compute SVD?