Bi7740: Scientific computing

Non-square linear systems

Vlad Popovici
popovici@iba.muni.cz

Institute of Biostatistics and Analyses
Masaryk University, Brno

Outline

1. Non-square systems
   - The underdetermined case
   - The overdetermined case
2. Numerical methods for LS problem
   - Orthogonal transformations
   - Singular Value Decomposition
   - Total least squares
3. Comparison of various decompositions

Outline

The systems of linear equations

General form:

$$\mathbf{Ax} = \mathbf{b}$$

$$\begin{bmatrix} a_{11} & \dots & a_{1n} \\ & \ddots & \\ a_{m1} & \dots & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix}$$

- if $m < n$: underdetermined case; find a minimum-norm solution
- if $m > n$: overdetermined case; minimize the squared error
- if $m = n$: determined case; already discussed

Reminder

- two vectors $\mathbf{y}, \mathbf{z}$ are orthogonal if $\mathbf{y}^T\mathbf{z} = 0$
- the span of a set of n independent vectors is $\text{span}(\{\mathbf{v}_1, \ldots, \mathbf{v}_n\}) = \left\{\sum_{i=1}^n \alpha_i \mathbf{v}_i \mid \alpha_i \in \mathbb{R}\right\}$
- the *row (column) space* of a matrix $\mathbf{A}$ is the linear subspace generated (or spanned) by the rows (colums) of $\mathbf{A}$. Its dimension is equal to $\text{rank}(\mathbf{A}) \leq \min(m, n)$.
- by definition, $\text{span}(\mathbf{A})$ is the column space of $\mathbf{A}$ and can be written as

$$C(\mathbf{A}) = \{\mathbf{v} \in \mathbb{R}^m : \mathbf{v} = \mathbf{A}\mathbf{x}, \mathbf{x} \in \mathbb{R}^n\},$$

so it is the space of transformed vectors by the action of multiplication by the matrix.

Outline

Underdetermined case

- $m < n$ there are more variables than equations, hence the solution is not unique
- consider the rows to be *linearly independent*
- then, any n-dimensional vector $\mathbf{x} \in \mathbb{R}^n$ can be decomposed into

$$\mathbf{x} = \mathbf{x}^+ + \mathbf{x}^-$$

where $\mathbf{x}^+$ is in the row space of $\mathbf{A}$ and $\mathbf{x}^-$ is in the null space of $\mathbf{A}$ (orthogonal to the previous space):

$$\mathbf{x}^+ = \mathbf{A}^T\alpha \qquad \mathbf{A}\mathbf{x}^- = 0$$

- this leads to

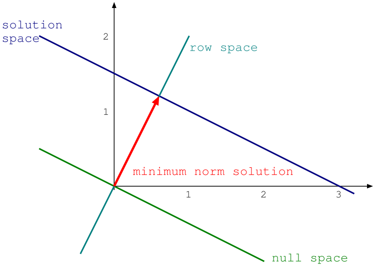
$$\mathbf{A}(\mathbf{x}^{+} + \mathbf{x}^{-}) = \mathbf{A}\mathbf{A}^{T}\alpha + \mathbf{A}\mathbf{x}^{-} = \mathbf{A}\mathbf{A}^{T}\alpha = \mathbf{b}$$

- $\mathbf{A}\mathbf{A}^{T}$ is a $m \times m$ nonsingular matrix, so $\mathbf{A}\mathbf{A}^{T}\alpha = \mathbf{b}$ has a unique solution $\alpha_0 = (\mathbf{A}\mathbf{A}^{T})^{-1}\mathbf{b}$
- the corresponding minimal norm solution to original system is

$$\mathbf{x}_0^{+} = \mathbf{A}^{T}(\mathbf{A}\mathbf{A}^{T})^{-1}\mathbf{b}$$

- note, however, that the orthogonal component $\mathbf{x}^{-}$ remains unspecified
- the matrix $\mathbf{A}^{T}(\mathbf{A}\mathbf{A}^{T})^{-1}$ is called the right pseudo-inverse of $\mathbf{A}$ (right: $\mathbf{A} \cdot \mathbf{A}^{T}(\mathbf{A}\mathbf{A}^{T})^{-1} = \mathbf{I}$)
- MATLAB: `pinv()`

Example: let $\mathbf{A} = [1\ 2]$ and $\mathbf{b} = [3]$ (hence $m = 1$).



- solution space:

$$x_2 = -\frac{1}{2}x_1 + \frac{3}{2}$$

is a solution, for any $x_1 \in \mathbb{R}$.

- $\mathbf{x}^+ = \mathbf{A}^T\alpha = \begin{bmatrix}1\\2\end{bmatrix}\alpha$ (row space)
- $\mathbf{Ax}^- = 0 \Rightarrow [1\ 2][x_1^-\ x_2^-]^T = 0.$
$\Rightarrow x_2^- = -\frac{1}{2}x_1^-$ (null space)

The minimal norm solution is *the intersection of solution space with the row space* and is the closest vector to the origin, among all vectors in the solution space:

$$\mathbf{x}_0^+ = [0.6\ 1.2]^T$$

Outline

1. Non-square systems
   - The underdetermined case
   - The overdetermined case
2. Numerical methods for LS problem
   - Orthogonal transformations
   - Singular Value Decomposition
   - Total least squares
3. Comparison of various decompositions

Overdetermined case

- if the rows of **A** are independent, there is no *perfect* solution to the system ($\mathbf{b} \notin \text{span}(\mathbf{A})$)
- one needs some other criterion to call a solution *acceptable*
- least squares solution $\mathbf{x}_0$ minimizes the square Euclidean norm of the residual vector:

$$\mathbf{x}_0 = \arg\min_{\mathbf{x}} ||\mathbf{r}||_2^2 = \arg\min_{\mathbf{x}} ||\mathbf{b} - \mathbf{A}\mathbf{x}||_2^2$$

Solution to the LS problem

From a linear system problem, we arrived at solving an optimization problem with objective function

$$J = \frac{1}{2}\|\mathbf{b} - \mathbf{A}\mathbf{x}\|_2^2 = \frac{1}{2}(\mathbf{b} - \mathbf{A}\mathbf{x})^T(\mathbf{b} - \mathbf{A}\mathbf{x})$$

Set the derivative wrt $\mathbf{x}$ to zero:

$$\frac{\partial}{\partial \mathbf{x}} J = \mathbf{A}^T\mathbf{b} - \mathbf{A}^T\mathbf{A}\mathbf{x} = 0$$

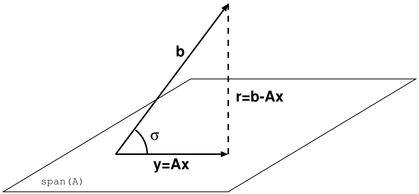
which leads to normal equations $\mathbf{A}^T\mathbf{A}\mathbf{x} = \mathbf{A}^T\mathbf{b}$, with the solution

$$\mathbf{x}_0 = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b}$$

$\mathbf{A}^\dagger = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T$ is the *left pseudo-inverse* of $\mathbf{A}$.

Solution to the LS problem - geometric interpretation

- let $\mathbf{y} = \mathbf{A}\mathbf{x}$, where $\mathbf{x}$ is the LS solution
- the residual $\mathbf{r} = \mathbf{b} - \mathbf{y}$ is orthogonal to span($\mathbf{A}$),



LS data approximation

Model: $y = c_3 x^2 - c_2 x + c_1$. Problem: $c_i = ?$ when (x_i, y_i) are given.

```
x = linspace(-2, 2, 100);
y = 2 * x.^2 - x + 3;                  % known model
yn = y + 2*(rand(1, 100) - 0.5);   % add some noise

A = [ones(100,1), x', x'.^2];          % Vandermonde matrix
coef = pinv(A) * y';             % solution, no noise in y
coefn = pinv(A) * yn';   % solution, uniform noise in y

hold on;
plot(x, yn, 'b.');
plot(x, coef(1) + coef(2) .* x + coef(3) .* x.^2, 'k-');
plot(x, coefn(1) + coefn(2) .* x + coefn(3) .* x.^2, ...
    'r-');
hold off;
```

Condition number

- if $\text{rank}(\mathbf{A}) = n$ (columns are independent), the condition number is
$$\text{cond}(\mathbf{A}) = ||\mathbf{A}||_2 ||\mathbf{A}^\dagger||_2$$
- by convention, if $\text{rank}(\mathbf{A}) < n$, $\text{cond}(\mathbf{A}) = \infty$
- for non-square matrices, the condition number measures the closeness to rank deficiency

Outline

Numerical methods for LS problem

- the LS solution can be obtained using the pseudo-inverse $\mathbf{A}^{\dagger} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T$ or by solving the normal equations

$$\mathbf{A}^T\mathbf{A}\mathbf{x} = \mathbf{A}^T\mathbf{b}$$

  which is a system of n equations
- $\mathbf{A}^T\mathbf{A}$ is symmetric positive definite, so it admits a Cholesky decomposition,

$$\mathbf{A}^T\mathbf{A} = \mathbf{L}\mathbf{L}^T$$

Issues with normal equations method

- floating-point computations in $\mathbf{A}^T\mathbf{A}$ and $\mathbf{A}^T\mathbf{b}$ may lead to information loss
- sensitivity of the solution is worsen, since $\text{cond}(\mathbf{A}^T\mathbf{A}) = [\text{cond}(\mathbf{A})]^2$

Example:

Let $\mathbf{A} = \begin{bmatrix} 1 & 1 \\ \epsilon & 0 \\ 0 & \epsilon \end{bmatrix}$ with $\epsilon \in \mathbb{R}_+$ and $\epsilon < \sqrt{\epsilon_{\text{mach}}}$. Then, in floating-point arithmetic, $\mathbf{A}^T\mathbf{A} = \begin{bmatrix} 1+\epsilon^2 & 1 \\ 1 & 1+\epsilon^2 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$ which is singular!

Augmented systems

- idea: find the solution and the residual as a solution of an extended system, under the orthogonality requirement
- the new system is

$$\begin{bmatrix} \mathbf{I} & \mathbf{A} \\ \mathbf{A}^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{r} \\ \mathbf{x} \end{bmatrix} = \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix}$$

- despite requiring more storage and not being positive definite, it allows more freedom in choosing pivots for LU decomposition
- in some cases it is useful, but not much used in practice

Outline

Orthogonal transformations

- a matrix $\mathbf{Q}$ is orthogonal if $\mathbf{Q}^T\mathbf{Q} = \mathbf{I}$
- multiplication of a vector by an orthogonal matrix does not change its *Euclidean* norm:

$$||\mathbf{Qv}||_2^2 = (\mathbf{Qv})^T\mathbf{Qv} = \mathbf{v}^T\mathbf{Q}^T\mathbf{Qv} = \mathbf{v}^T\mathbf{v} = ||\mathbf{v}||_2^2$$

- so, multiplying the two sides of the system by $\mathbf{Q}$ does not change the solution
- again: try to transform the system so it's easy to solve e.g. triangular system

- an upper triangular overdetermined ($m > n$) LS problem has the form
$$\begin{bmatrix}\mathbf{R}\\ \mathbf{0}\end{bmatrix}\mathbf{x} \approx \begin{bmatrix}\mathbf{b}_1\\ \mathbf{b}_2\end{bmatrix}$$
where $\mathbf{R}$ is an $n \times n$ upper triangular matrix and $\mathbf{b}$ is partitioned accordingly
- the residual becomes
$$||\mathbf{r}||_2^2 = ||\mathbf{b}_1 - \mathbf{R}\mathbf{x}||_2^2 + ||\mathbf{b}_2||_2^2$$
- to minimize the residual, one has to minimize $||\mathbf{b}_1 - \mathbf{R}\mathbf{x}||_2^2$ (since $||\mathbf{b}_2||_2^2$ is fixed) and this leads to the system
$$\mathbf{R}\mathbf{x} = \mathbf{b}_1$$
which can be solved by back-substitution
- the residual becomes $||\mathbf{r}||_2^2 = ||\mathbf{b}_2||_2^2$ and $\mathbf{x}$ is the LS solution

QR factorization

- problem: find an $m \times m$ orthogonal matrix $\mathbf{Q}$ such that an $m \times n$ matrix $\mathbf{A}$ can be written as

$$\mathbf{A} = \mathbf{Q} \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix}$$

where $\mathbf{R}$ is $n \times n$ upper triangular

- the new problem to solve is

$$\mathbf{Q}^T \mathbf{A} \mathbf{x} = \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix} \mathbf{x} \approx \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{bmatrix} = \mathbf{Q}^T \mathbf{b}$$

- if $\mathbf{Q}$ is partitioned as $\mathbf{Q} = [\mathbf{Q}_1\mathbf{Q}_2]$ with $\mathbf{Q}_1$ having n columns, then

$$\mathbf{A} = \mathbf{Q}\begin{bmatrix}\mathbf{R} \\ \mathbf{0}\end{bmatrix} = \mathbf{Q}_1\mathbf{R}$$

  is called reduced QR factorization of $\mathbf{A}$ (MATLAB: `[Q,R] = qr(A, 0)`)
- columns of $\mathbf{Q}_1$ form an orthonormal basis of span($\mathbf{A}$), and the columns of $\mathbf{Q}_2$ forn an orthonormal basis of span$(\mathbf{A})^\perp$
- $\mathbf{Q}_1\mathbf{Q}_1^T$ is orthogonal projector onto span($\mathbf{A}$)
- the solution to the initial problem is given by the solution to the square system

$$\mathbf{Q}_1^T\mathbf{A}\mathbf{x} = \mathbf{Q}_1^T\mathbf{b}$$

QR factorization - Remember:

In general, for an $m \times n$ matrix A, with $m > n$, the factorization is

$$\mathbf{A} = \mathbf{QR}$$

and

- $\mathbf{Q}$ is an *orthogonal* matrix: $\mathbf{Q}^T\mathbf{Q} = \mathbf{I} \Leftrightarrow \mathbf{Q}^{-1} = \mathbf{Q}^T$
- $\mathbf{R}$ is an *upper triangular* matrix
- solving the normal equations (for LS solution) $\mathbf{A}^T\mathbf{A}\mathbf{x} = \mathbf{A}^T\mathbf{b}$ comes to solving

$$\mathbf{R}\mathbf{x} = \mathbf{Q}^T\mathbf{b}$$

HOMEWORK: prove the above statement.

Example

```
A = [3 -6; 4 -8; 0 1];
b = [-1 7 2]';
[Q1,R] = qr(A,0)
d = Q1*b;
bksolve(R, d)    % remember back-subsitution method

% equivalent (linear regression!):
regress(b, A)
```

A statistical perspective

Changing a bit the notation, the linear model is

$$E[\mathbf{y}] = \mathbf{X}\beta, \qquad \mathrm{Cov}(y) = \sigma^2 I$$

It can be shown that the *best linear unbiased estimator* is

$$\hat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} = \mathbf{R}^{-1}\mathbf{Q}^T\mathbf{y}$$

for a decomposition $\mathbf{X} = \mathbf{QR}$. Then $\hat{\mathbf{y}} = \mathbf{QQ}^T\mathbf{y}$. (Gauss-Markov thm.: LS estimator has the lowest variance among all unbiased linear estimators.) Also,

$$\mathrm{Var}(\hat{\beta}) = (\mathbf{X}^T\mathbf{X})^{-1}\sigma^2 = (\mathbf{R}^T\mathbf{R})^{-1}\sigma^2$$

where $\sigma^2 = \|\mathbf{y} - \hat{\mathbf{y}}\|^2/(m - n - 1)$.

Computing the QR factorization

- similarly to LU factorization, we nullify entries under the diagonal, column by column
- now, use orthogonal transformations:
  - Householder transformations
  - Givens rotations
  - Gram-Schmidt orthogonalization
- MATLAB:`qr()`

Householder transformations

$$\mathbf{H} = \mathbf{I} - 2\frac{\mathbf{v}\mathbf{v}^T}{\mathbf{v}^T\mathbf{v}}, \qquad \mathbf{v} \neq 0$$

- $\mathbf{H}$ is orthogonal and symmetric: $\mathbf{H} = \mathbf{H}^T = \mathbf{H}^{-1}$
- $\mathbf{v}$ are chosen such that for a vector $\mathbf{a}$:

$$\mathbf{H}\mathbf{a} = \begin{bmatrix} \alpha \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \alpha \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \alpha \mathbf{e}_1$$

- this leads to $\mathbf{v} = \mathbf{a} - \alpha\mathbf{e}_1$ with $\alpha = \pm||\mathbf{a}||_2$, where the sign is chosen to avoid cancellation

Householder QR factorization

- apply, the Householder transformation to nuliffy the entries below diagonal
- the process is applied to each column (of the n) and produces a transformation of the form

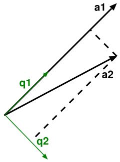
$$\mathbf{H}_n \ldots \mathbf{H}_1 \mathbf{A} = \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix}$$

  where $\mathbf{R}$ is $n \times n$ upper triangular
- then take $\mathbf{Q} = \mathbf{H}_1 \ldots \mathbf{H}_n$
- note that the multiplication of $\mathbf{H}$ with a vector $\mathbf{u}$ is much cheaper than a general matrix-vector multiplication:

$$\mathbf{H}\mathbf{u} = \left(\mathbf{I} - 2\frac{\mathbf{v}\mathbf{v}^T}{\mathbf{v}^T\mathbf{v}}\right)\mathbf{u} = \mathbf{u} - 2\frac{\mathbf{v}^T\mathbf{u}}{\mathbf{v}^T\mathbf{v}}\mathbf{v}$$

Gram-Schmidt orthogonalization

- idea: given two vectors $\mathbf{a}_1$ and $\mathbf{a}_2$, we seek orthonormal vectors $\mathbf{q}_1$ and $\mathbf{q}_2$ having the same span



- method: subtract from $\mathbf{a}_2$ its projection on $\mathbf{a}_1$ and normalize the resulting vectors
- apply this method to each column of $\mathbf{A}$ to obtain the classical Gram-Schmidt procedure

Algorithm: Classical Gram-Schmidt

for $k = 1$ **to** n **do**
  $\mathbf{q}_k \leftarrow \mathbf{a}_k$;
  for $j = 1$ **to** $k - 1$ **do**
    $r_{jk} \leftarrow \mathbf{q}_j^T \mathbf{a}_k$;
    $\mathbf{q}_k \leftarrow \mathbf{q}_k - r_{jk}\mathbf{q}_j$;
  end for
  $r_{kk} \leftarrow \|\mathbf{q}_k\|_2$;
  $\mathbf{q}_k \leftarrow \mathbf{q}_k / r_{kk}$;
end for

The resulting matrices $\mathbf{Q}$ (with $\mathbf{q}_k$ as columns) and $\mathbf{R}$ (with elements r_{jk}) form the reduced QR factorization of $\mathbf{A}$.

Modified Gram-Schmidt procedure

- improved orthogonality in finite-precision
- reduced storage requirements
- HOMEWORK: implement the procedure below in Matlab

Algorithm: Modified Gram-Schmidt

for $k = 1$ **to** n **do**
  $r_{kk} \leftarrow \|\mathbf{a}_k\|_2$;
  $\mathbf{q}_k \leftarrow \mathbf{a}_k / r_{kk}$;
  for $j = k+1$ **to** n **do**
    $r_{jk} \leftarrow \mathbf{q}_k^T \mathbf{a}_j$;
    $\mathbf{a}_j \leftarrow \mathbf{a}_j - r_{kj}\mathbf{q}_k$;
  end for
end for

Further topics on QR factorization

- if $\text{rank}(\mathbf{A}) < n$ then $\mathbf{R}$ is singular and there are multiple solutions $\mathbf{x}$; choose the $\mathbf{x}$ with the smallest norm
- in limited precision, the rank can be lower than the theoretical one, leading to highly sensitive solutions → an alternative could be the SVD method (later)
- there exists a version, QR with pivoting, that chooses everytime the column with largest Euclidean norm for reduction → improves stability in rank deficient scenarios
- another method of factorization: Givens rotations - makes one 0 at a time

Outline

Singular Value Decomposition - SVD

- SVD of an $m \times n$ matrix $\mathbf{A}$ has the form

$$\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^T$$

  where $\mathbf{U}$ is $m \times m$ orthogonal matrix, $\mathbf{V}$ is $n \times n$ orthogonal matrix, and Σ is $m \times n$ diagonal matrix, with

$$\sigma_{ij} = \begin{cases} 0 & \text{if } i \neq j \\ \sigma_i \geq 0 & \text{if } i = j \end{cases}$$

- σ_i are usually ordered such that $\sigma_1 \geq \cdots \geq \sigma_n$ and are called singular values of $\mathbf{A}$
- the columns $\mathbf{u}_i$ and $\mathbf{v}_i$ are called left and right singular vectors of $\mathbf{A}$, respectively

- minimum norm solution to $\mathbf{Ax} \approx \mathbf{b}$ is

$$\mathbf{x} = \sum_{\sigma_i \neq 0} \frac{\mathbf{u}_i^T \mathbf{b}}{\sigma_i} \mathbf{v}_i$$

- for ill-conditioned or rank-deficient problems, the sum should be taken over "large enough" σ's: $\sum_{\sigma_i \geq \epsilon} \dots$
- Euclidean norm: $||\mathbf{A}||_2 = \max_i\{\sigma_i\}$
- Euclidean condition number: $\mathrm{cond}(\mathbf{A}) = \frac{\max_i\{\sigma_i\}}{\min_i\{\sigma_i\}}$
- Rank of $\mathbf{A}$: $\mathrm{rank}(\mathbf{A}) = \#\{\sigma_i > 0\}$

Pseudoinverse (again)

- the pseudoinverse of an $m \times n$ matrix $\mathbf{A}$ with SVD decomposition $\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^T$ is

$$\mathbf{A}^+ = \mathbf{V}\Sigma^{-1}\mathbf{U}^T$$

where

$$[\Sigma^{-1}]_{ii} = \begin{cases} 1/\sigma_i & \text{for } \sigma_i > 0 \\ 0 & \text{otherwise} \end{cases}$$

- pseudoinverse always exists and minimum norm solution to $\mathbf{Ax} \approx \mathbf{b}$ is $\mathbf{x} = \mathbf{A}^+\mathbf{b}$
- if $\mathbf{A}$ is square and nonsingular, $\mathbf{A}^{-1} = \mathbf{A}^+$

SVD and subspaces relevant to $\mathbf{A}$

- $\mathbf{u}_i$ for which $\sigma_i > 0$ form the orthonormal basis of span($\mathbf{A}$)
- $\mathbf{u}_i$ for which $\sigma_i = 0$ form the orthonormal basis of the orthogonal complement of span($\mathbf{A}$)
- $\mathbf{v}_i$ for which $\sigma_i = 0$ form the orthonormal basis of the null space of $\mathbf{A}$
- $\mathbf{v}_i$ for which $\sigma_i > 0$ form the orthonormal basis of the orthogonal complement of the null space of $\mathbf{A}$

SVD and matrix approximation

- $\mathbf{A}$ can be re-written as

$$\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^T = \sigma_1 \mathbf{u}_1 \mathbf{v}_1^T + \cdots + \sigma_n \mathbf{u}_n \mathbf{v}_n^T$$

- let $\mathbf{E}_i = \mathbf{u}_i \mathbf{v}_i^T$; $\mathbf{E}_i$ has rank 1 and requires only $m + n$ storage locations
- $\mathbf{E}_i \mathbf{x}$ multiplication requires only $m + n$ multiplications
- assuming $\sigma_1 \geq \sigma_2 \geq \ldots \sigma_n$ then by using the largest k singular values, one obtains the closes approximation of $\mathbf{A}$ of rank k:

$$\mathbf{A} \approx \sum_{i=1}^{k} \sigma_i \mathbf{E}_i$$

- many applications to image processing, data compression, cryptography, etc.

Example - image compression

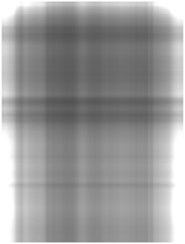
MATLAB: `[U,S,V] = svd(X)`

Original image and its approximations using 1,2,3,4,5 and 10 terms:

Example - image compression

MATLAB: `[U,S,V] = svd(X)`

Original image and its approximations using 1,2,3,4,5 and 10 terms:



Example - image compression

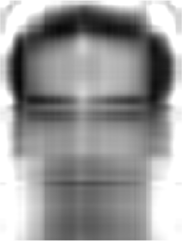
MATLAB: `[U,S,V] = svd(X)`

Original image and its approximations using 1,2,3,4,5 and 10 terms:

Example - image compression

MATLAB: `[U,S,V] = svd(X)`

Original image and its approximations using 1,2,3,4,5 and 10 terms:



Example - image compression

MATLAB: `[U,S,V] = svd(X)`

Original image and its approximations using 1,2,3,4,5 and 10 terms:

Example - image compression

MATLAB: `[U,S,V] = svd(X)`

Original image and its approximations using 1,2,3,4,5 and 10 terms:

Example - image compression

MATLAB: `[U,S,V] = svd(X)`

Original image and its approximations using 1,2,3,4,5 and 10 terms:

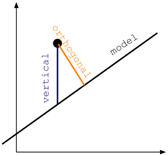
```
X = imread('face01.png');
X = double(X) ./ 255.0;

[U,S,V] = svd(X);

imshow(U(:,1)  * S(1,1) * V(:,1)' );
imshow(U(:,1:5)* S(1:5, 1:5) * V(:,1:5)');
imshow(U(:,1:10)* S(1:10, 1:10) * V(:,1:10)');
```

Outline

Total least squares

$$\mathbf{Ax} \approxeq \mathbf{b}$$

- ordinary least squares applies when the error affects only **b**
- what if there is error (uncertainty) in **A** as well?
- total least squares minimizes the orthogonal distances, rather than vertical distances, between model and data



- can be computed using SVD of $[\mathbf{A}, \mathbf{b}]$

Outline

Comparison: work effort

- computing $\mathbf{A}^T\mathbf{A}$ requires about $n^2m/2$ multiplications and solving the resulting symmetric system, about $n^3/6$ multiplications
- LS problem solution by Householder QR requires about $mn^2 - n^3/3$ multiplications
- if $m \gg n$, Householder method requires about twice as much work normal eqs.
- cost of SVD is $\approx (4 \ldots 10) \times (mn^2 + n^3)$ depending on implementation

Comparison: precision

- relative error for normal eqs. is $\sim [\text{cond}(\mathbf{A})]^2$; if $\text{cond}(\mathbf{A}) \approx 1/\sqrt{\epsilon_{\text{mach}}}$, Cholesky factorization will break
- Householder method has a relative error

$$\sim \text{cond}(\mathbf{A}) + \|\mathbf{r}\|_2 [\text{cond}(\mathbf{A})]^2$$

  which is the best achievable for LS problems
- Householder method breaks (in back-substitution step) for $\text{cond}(\mathbf{A}) \lessapprox 1/\epsilon_{\text{mach}}$
- while Householder method is more general and more accurate than normal equations, it may not always be worth the additional cost

Comparison: precision, cont'd

- for (nearly) rank-deficient problems, the pivoting Householder method produces useful solution, while normal equations method fails
- SVD is more precise and more robust than Householder method, but much more expensive computationally