STAT 309: MATHEMATICAL COMPUTATIONS I
FALL 2015
LECTURE 7

1. Finding closest unitary/orthogonal matrix

- let $U(n)$ be the set of all $n \times n$ unitary matrices
- given $A \in \mathbb{C}^{n\times n}$, we wish to find the matrix $X \in U(n)$ that satisfies
$$\min_{X\in U(n)} \|A - X\|_F$$
- let $A = U\Sigma V^*$ be the SVD of A
- if we set
$$X = UV^*,$$
then
$$\|A - X\|_F^2 = \|U(\Sigma - I)V^*\|_F^2 = \|\Sigma - I\|_F^2 = (\sigma_1 - 1)^2 + \cdots + (\sigma_n - 1)^2$$
- it can be shown that this is in fact the minimum (see Homework 2)
- for real matrices A, one could also ask for
$$\min_{X\in O(n)} \|A - X\|_F$$
which is just a special case

2. Procrustes problem

- a more general problem is to find $X \in U(n)$ such that
$$\min_{X\in U(n)} \|A - BX\|_F$$
for given matrices $A, B \in \mathbb{C}^{m\times n}$
- let $B^*A = U\Sigma V^*$ be the SVD of B^*A
- the solution is given by
$$X = UV^*$$
- you will be asked to prove this in Homework 2

3. Aside: closest Hermitian/symmetric matrix

- this one doesn't require SVD but is interesting nonetheless
- given $A \in \mathbb{C}^{n\times n}$, find its closest Hermitian matrix
$$\min_{X^*=X} \|A - X\|_F \tag{3.1}$$
or its closest skew-Hermitian matrix
$$\min_{X^*=-X} \|A - X\|_F \tag{3.2}$$
- note that any square matrix can be written as a sum of a Hermitian matrix and a skew-Hermitian matrix
$$A = \frac{1}{2}(A + A^*) + \frac{1}{2}(A - A^*)$$

Date: October 28, 2015, version 1.0. Comments, bug reports: `lekheng@galton.uchicago.edu`.

- the solutions to (3.1) and (3.2) are given by $X = \frac{1}{2}(A+A^*)$ and $X = \frac{1}{2}(A-A^*)$ respectively (why?)
- for $A \in \mathbb{R}^{n\times n}$ these yield the closest symmetric and skew-symmetric matrices to A

4. Finding a best rank-r approximation

- given $A \in \mathbb{C}^{m\times n}$, we want to find $X \in \mathbb{C}^{m\times n}$ of rank not more than r so that $\|A - X\|$ is minimized
- in notations, we want
$$\min_{\operatorname{rank}(X)\le r} \|A - X\| \tag{4.1}$$
- such an X is called a best rank-r approximation to A or a rank-r projection of A
- if $r \ge \operatorname{rank}(A)$, then clearly $X = A$ and the problem is trivial
- so we shall always assume that $r < \operatorname{rank}(A)$
- we will see how to construct such an X explicitly when the norm $\|\cdot\|$ is unitarily invariant, i.e., satisfying
$$\|UXV\| = \|X\|$$
for all $X \in \mathbb{C}^{m\times n}$ whenever U and V are unitary matrices
- we will start with the classical case where $\|\cdot\|$ is the matrix 2-norm or spectral norm

Theorem 1 (Eckart–Young)**.** *Let the* SVD *of A be*
$$A = \sum_{i=1}^{\operatorname{rank}(A)} \sigma_i \mathbf{u}_i \mathbf{v}_i^*, \quad \sigma_1 \ge \cdots \ge \sigma_r > 0.$$
Then for any $r \in \{1, \dots, \operatorname{rank}(A) - 1\}$, a solution to (4.1) when $\|\cdot\| = \|\cdot\|_2$ is given by
$$X = \sum_{i=1}^{r} \sigma_i \mathbf{u}_i \mathbf{v}_i^*.$$
Furthermore, we have
$$\min_{\operatorname{rank}(X)\le r} \|A - X\|_2 = \sigma_{r+1}. \tag{4.2}$$
In matrix form, we have
$$X = U \begin{bmatrix} \sigma_1 & & & & & \\ & \ddots & & & & \\ & & \sigma_r & & & \\ & & & 0 & & \\ & & & & \ddots & \\ & & & & & 0 \end{bmatrix} V^*, \tag{4.3}$$
where $A = U\Sigma V^*$ is the SVD *of A.*

Proof. Suppose there is a $B \in \mathbb{C}^{m\times n}$ with $\operatorname{rank}(B) \le r$ and $\|A - B\|_2 < \sigma_{r+1}$. Then by the rank-nullity theorem
$$\operatorname{rank}(B) + \dim(\ker(B)) = n$$
and so
$$\dim(\ker(B)) \ge n - r.$$
Let $\mathbf{w} \in \ker(B)$. Then $B\mathbf{w} = \mathbf{0}$ and so
$$\|A\mathbf{w}\|_2 = \|(A - B)\mathbf{w}\|_2 \le \|A - B\|_2 \|\mathbf{w}\|_2 < \sigma_{r+1}\|\mathbf{w}\|_2. \tag{4.4}$$

Let $\mathbf{w} \in W := \operatorname{span}\{\mathbf{v}_1, \dots, \mathbf{v}_{r+1}\}$. Then $\mathbf{w} = \alpha_1 \mathbf{v}_1 + \cdots + \alpha_{r+1}\mathbf{v}_{r+1}$. Rewriting this in matrix form

$$\mathbf{w} = V_{r+1} \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_{r+1} \end{bmatrix} = V_{r+1}\boldsymbol{\alpha}$$

where $V_{r+1} = [\mathbf{v}_1, \dots, \mathbf{v}_{r+1}] \in \mathbb{C}^{n \times r}$, i.e., the first $r+1$ columns of V.

$$\|A\mathbf{w}\|_2^2 = \|U\Sigma V^* V_{r+1}\boldsymbol{\alpha}\|_2^2 = \left\| \Sigma \begin{bmatrix} I_{r+1} \\ O \end{bmatrix} \boldsymbol{\alpha} \right\|_2^2 = \left\| \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_{r+1} \\ 0 & \cdots & 0 \\ \vdots & & \vdots \\ 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_{r+1} \end{bmatrix} \right\|_2^2$$

$$= \sum_{i=1}^{r+1} \sigma_i^2 |\alpha_i|^2 \geq \sigma_{r+1}^2 \sum_{i=1}^{r+1} |\alpha_i|^2 = \sigma_{r+1}^2 \|\mathbf{w}\|_2^2.$$

Hence if $\mathbf{w} \in W$, then

$$\|A\mathbf{w}\|_2 \geq \sigma_{r+1}\|\mathbf{w}\|_2. \tag{4.5}$$

But since $\dim(\ker(B)) \geq n - r$ and $\dim(W) = r + 1$, the two subspaces must intersect nontrivially, i.e., $\dim(\ker(B) \cap W)) \geq 1$ and so there exists a non-zero vector $\mathbf{w} \in \ker(B) \cap W$. Such a vector would satisfy both (4.4) and (4.5), a contradiction. Hence our original assumption is false: There is no rank-r matrix B that could beat the bound in (4.2). On the other hand it is easy to verify that the choice of X in (4.3) satisfies (4.2):

$$\|A - X\|_2 = \left\| U \begin{bmatrix} 0 & & & & & & & \\ & \ddots & & & & & & \\ & & 0 & & & & & \\ & & & \sigma_{r+1} & & & & \\ & & & & \ddots & & & \\ & & & & & \sigma_{\operatorname{rank}(A)} & & \\ & & & & & & 0 & \\ & & & & & & & \ddots & \\ & & & & & & & & 0 \end{bmatrix} V^* \right\|_2 = \sigma_{r+1}.$$

□

- the generalization of Eckart–Young theoem to any arbitrary unitarily invariant norm is due to Mirsky and this theorem is sometimes also called the Eckart–Young–Mirsky theorem
- note that the general theorem only says that (5.2) is the best rank-r approximation of A, the value in (4.2) would in general be different
- for example if we use the Frobenius norm

$$\min_{\operatorname{rank}(X) \leq r} \|A - X\|_F = \sqrt{\sigma_{r+1}^2 + \cdots + \sigma_{\operatorname{rank}(A)}^2}.$$

5. Computing condition number

- we defined the 2-norm condition number for a nonsingular square matrix $A \in \mathbb{C}^{n \times n}$ as

$$\kappa_2(A) = \|A\|_2 \|A^{-1}\|_2 \tag{5.1}$$

- what if A is singular? one way is to set $\kappa_2(A) = \infty$

- this is natural not very useful — the only information it convey is what you already know, namely, A is singular
- if we apply SVD of A and the unitary invariance of the 2-norm, then an alternative expression for (5.1) is

$$\kappa_2(A) = \frac{\sigma_1(A)}{\sigma_n(A)} = \frac{\sigma_{\max}(A)}{\sigma_{\min}(A)}$$

- note that $\operatorname{rank}(A) = n$ and we could have written

$$\kappa_2(A) = \frac{\sigma_1(A)}{\sigma_{\operatorname{rank}(A)}(A)} = \frac{\sigma_{\max}(A)}{\sigma_{\min}(A)} \tag{5.2}$$

  where $\sigma_{\min}(A)$ denotes the smallest non-zero singular value of A
- this last expression extends to any singular and even rectangular $A \in \mathbb{C}^{m\times n}$ as long as $A \neq O$
- note that $\sigma_{\operatorname{rank}(A)}(A)$ is the smallest non-zero singular value of A
- we call (5.2) the *generalized condition number* to distinguish it from (5.1)
- another expression for (5.2) is

$$\kappa_2(A) = \|A\|_2\|A^\dagger\|_2 \tag{5.3}$$

- proof: use SVD to see that $\|A\|_2 = \sigma_1(A)$ and $\|A^\dagger\|_2 = \sigma_{\operatorname{rank}(A)}(A)$
- (5.3) can be used to extend generalized condition number to any matrix norm, for example

$$\kappa_p(A) = \|A\|_p\|A^\dagger\|_p, \quad \kappa_F(A) = \|A\|_F\|A^\dagger\|_F$$

6. LEAST SQUARES WITH QUADRATIC CONSTRAINTS

- let $A \in \mathbb{R}^{m\times n}$, $\mathbf{b} \in \mathbb{R}^m$, and α be some given positive number
- we wish to solve the problem

$$\begin{aligned} \text{minimize} \quad & \|\mathbf{b} - A\mathbf{x}\|_2 \\ \text{subject to} \quad & \|\mathbf{x}\|_2 \le \alpha \end{aligned} \tag{6.1}$$

- this problem is known as *least squares with quadratic constraints*
- arises in many situations:
  - ridge regression
  - Tychonov regularization
  - generalized cross-validation (GCV)
- note that if $\alpha \ge \|A^\dagger \mathbf{b}\|_2$, the unconstrained minimum norm solution $A^\dagger\mathbf{b}$ would already be a solution
- so for a non-trivial solution, we assume that $\alpha < \|A^\dagger\mathbf{b}\|_2$ and in which case the solution $\mathbf{x}$ to (6.1) must sit on the boundary of the ball of radius α, i.e., $\|\mathbf{x}\|_2 = \alpha$
- to solve this problem, we define the *Lagrangian*

$$L(\mathbf{x}, \mu) = \|\mathbf{b} - A\mathbf{x}\|_2^2 + \mu(\|\mathbf{x}\|^2 - \alpha^2)$$

  where μ is called the *Lagrange multiplier*
- first-order condition for minimality: set derivative to zero

$$\mathbf{0} = \nabla_{\mathbf{x}} L(\mathbf{x}, \mu) = -2A^\mathsf{T}\mathbf{b} + 2A^\mathsf{T}A\mathbf{x} + 2\mu\mathbf{x}$$

- we obtain

$$(A^\mathsf{T}A + \mu I)\mathbf{x} = A^\mathsf{T}\mathbf{b} \tag{6.2}$$

- if we denote the eigenvalues of $A^\mathsf{T}A$ by

$$\lambda_1 \ge \lambda_2 \ge \cdots \ge \lambda_n \ge 0$$

- then eigenvalues of $A^\mathsf{T}A + \mu I$ are
$$\lambda_1 + \mu, \cdots, \lambda_n + \mu$$
- if $\mu \geq 0$, then $\kappa_2(A^\mathsf{T}A + \mu I) \leq \kappa_2(A^\mathsf{T}A)$, because
$$\frac{\lambda_1 + \mu}{\lambda_n + \mu} \leq \frac{\lambda_1}{\lambda_n}$$
- so $A^\mathsf{T}A + \mu I$ is better conditioned
- to solve (6.2), we see that we need to compute
$$\mathbf{x} = (A^\mathsf{T}A + \mu I)^{-1}A^\mathsf{T}\mathbf{b} \tag{6.3}$$
where
$$\mathbf{x}^\mathsf{T}\mathbf{x} = \mathbf{b}^\mathsf{T}A(A^\mathsf{T}A + \mu I)^{-2}A^\mathsf{T}\mathbf{b} = \alpha^2$$
- if $A = U\Sigma V^\mathsf{T}$ is the full SVD of A, we let $\mathbf{c} = U^\mathsf{T}\mathbf{b}$, then we have
$$\begin{aligned}
\alpha^2 &= \mathbf{b}^\mathsf{T}U\Sigma V^\mathsf{T}(V\Sigma^\mathsf{T}\Sigma V^\mathsf{T} + \mu I)^{-2}V\Sigma^\mathsf{T}U^\mathsf{T}\mathbf{b} \\
&= \mathbf{c}^\mathsf{T}\Sigma[(V\Sigma^\mathsf{T}\Sigma V^\mathsf{T} + \mu I)V]^{-1}[V^\mathsf{T}(V\Sigma^\mathsf{T}\Sigma V^\mathsf{T} + \mu I)]^{-1}\Sigma^\mathsf{T}\mathbf{c} \\
&= \mathbf{c}^\mathsf{T}\Sigma(V\Sigma^\mathsf{T}\Sigma + \mu V)^{-1}(\Sigma^\mathsf{T}\Sigma V^\mathsf{T} + \mu V^\mathsf{T})^{-1}\Sigma^\mathsf{T}\mathbf{c} \\
&= \mathbf{c}^\mathsf{T}\Sigma[(\Sigma^\mathsf{T}\Sigma V^\mathsf{T} + \mu V^\mathsf{T})(V\Sigma^\mathsf{T}\Sigma + \mu V)]^{-1}\Sigma^\mathsf{T}\mathbf{c} \\
&= \mathbf{c}^\mathsf{T}\Sigma(\Sigma^\mathsf{T}\Sigma + \mu I)^{-2}\Sigma^\mathsf{T}\mathbf{c} \\
&= \sum_{i=1}^{r} \frac{c_i^2\sigma_i^2}{(\sigma_i^2 + \mu)^2} \\
&=: f(\mu)
\end{aligned}$$
where $\mathbf{c} = (c_1, \dots, c_m)^\mathsf{T}$
- the function $f(\mu)$ has poles at $-\sigma_i^2$ for $i = 1, \dots, r$
- furthermore, $\lim_{\mu\to\infty} f(\mu) = 0$
- algorithm for solving this problem, given A, $\mathbf{b}$, and α^2:
  - step 1: compute SVD of A to obtain $A = U\Sigma V^\mathsf{T}$
  - step 2: compute $\mathbf{c} = U^\mathsf{T}\mathbf{b}$
  - step 3: solve $f(\mu_*) = \alpha^2$ with Newton–Raphson method
  - step 4: use the SVD to compute
$$\mathbf{x} = (A^\mathsf{T}A + \mu I)^{-1}A^\mathsf{T}\mathbf{b} = V(\Sigma^\mathsf{T}\Sigma + \mu I)^{-1}\Sigma^\mathsf{T}U^\mathsf{T}\mathbf{b}$$
- don't use Newton–Raphson method on this equation directly; solving $1/f(\mu) = 1/\alpha^2$ is much better
- this is an example of an 'almost closed form' solution: we have an analytic expression for $\mathbf{x}$ that depends on just one unknown parameter μ_*, which is the root of a univariate nonlinear equation

7. Aside: why orthogonal/unitary

- unitary and orthogonal matrices are awesome because they preserve length
- it also preserves the length of your errors and so your errors don't get magnified during your computations
- more precisely, if we multiply a vector $\mathbf{a} \in \mathbb{C}^n$ or a matrix $A \in \mathbb{C}^{n\times k}$ by another matrix $X \in \mathrm{GL}(n)$ we usually magnify whatever error there is in $\mathbf{a}$ or A by $\kappa_2(X)$, the condition number of X
- more precisely, unitary and orthogonal matrices are awesome because they are perfectly conditioned, i.e., $\kappa_2(U) = 1$ for all $U \in U(n)$

- the vector case $\mathbf{a} \in \mathbb{C}^n$ is the same as the matrix case $A \in \mathbb{C}^{n \times k}$ with $k = 1$ so we will do the more general one
- for simplicity, let us assume that we know X precisely but
  - we don't have XA, only $\mathrm{fl}(XA)$, which differs from XA by an error term E

$$\mathrm{fl}(XA) = XA + E$$

  - we have also assumed that all errors arise from rounding in floating point arithmetic and storage
- we will do something called *backward error analysis*, i.e., we want to find the smallest perturbation ΔA in A so that $XA + E$ is the *exact* answer had $A + \Delta A$ been the input
- we will measure how good our method is by asking what is the relative error in the input

$$\frac{\|\Delta A\|_2}{\|A\|_2} \tag{7.1}$$

  required so that the relative error of the output is

$$\frac{\|E\|_2}{\|XA\|_2} \leq \varepsilon \tag{7.2}$$

  for some $\varepsilon > 0$
- terminologies: the ratio in (7.1) is called the relative backward error, the ratio in (4.1) is called the relative forward error
- in this case, it is trivial to derive the relative backward error: by assumption $XA + E$ is the *exact* answer of multiplying X to $A + \Delta A$, so

$$XA + E = X(A + \Delta A)$$

  and so

$$\Delta A = X^{-1}E$$

- from (7.2), we get $\|E\|_2 \leq \varepsilon\|XA\|_2 \leq \varepsilon\|X\|_2\|A\|_2$ and so

$$\|\Delta A\|_2 \leq \|X^{-1}\|_2\|E\|_2 \leq \varepsilon\kappa_2(X)\|A\|_2$$

  and so the relative backward error is

$$\frac{\|\Delta A\|_2}{\|A\|_2} \leq \varepsilon\kappa_2(X) \tag{7.3}$$

- this may seem a little wierd the first time you see it: why don't we assume that the error is in the input and then see how big it becomes in the output — this is called forward error analysis
- forward error analysis is in general much hard than backward error analysis
- recap of backward error analysis
  - we assume that the error E in the final computed output comes from the *exact* solution of a perturbed problem $A + \Delta A$
  - we start by assuming that the relative error in the output is ε, i.e., (7.2)
  - then we try to find how far away (i.e., ΔA) the input must be from the given one (i.e., A) in order to produce such an error ε in the output, i.e., (7.3), when everything is done without error
- we will cover backward error analysis and condition number in greater details later

8. SOLVING TOTAL LEAST SQUARES PROBLEMS

- assume $A \in \mathbb{C}^{m\times n}$ has full column rank, i.e., $\operatorname{rank}(A) = n \le m$
- in ordinary least squares problem, we solve
$$A\mathbf{x} = \mathbf{b} + \mathbf{r}, \quad \|\mathbf{r}\|_2 = \min$$
- in *total least squares* problem, we wish to solve
$$(A+E)\mathbf{x} = \mathbf{b} + \mathbf{r}, \quad \|E\|_F^2 + \lambda^2\|\mathbf{r}\|_2^2 = \min$$
- from $A\mathbf{x} - \mathbf{b} + E\mathbf{x} - \mathbf{r} = \mathbf{0}$ we obtain the system
$$\begin{bmatrix} A & \mathbf{b} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ -1 \end{bmatrix} + \begin{bmatrix} E & \mathbf{r} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ -1 \end{bmatrix} = \mathbf{0}$$
or
$$(C+F)\mathbf{z} = \mathbf{0} \tag{8.1}$$
- since
$$\mathbf{z} = \begin{bmatrix} \mathbf{x} \\ -1 \end{bmatrix} \neq \mathbf{0} \tag{8.2}$$
we know that $\operatorname{nullity}(C+F) \ge 1$ and so $\operatorname{rank}(C+F) \le n$
- we need the matrix $C+F$ to have $\operatorname{rank}(C+F) \le n$, and we want to minimize $\|F\|_F$
- to solve this problem, we compute the SVD of $C \in \mathbb{C}^{m\times(n+1)}$
$$C = \begin{bmatrix} A & \mathbf{b} \end{bmatrix} = U\Sigma V^* = U \begin{bmatrix} \sigma_1 & & & \\ & \ddots & & \\ & & \sigma_n & \\ & & & \sigma_{n+1} \\ 0 & \cdots & \cdots & 0 \\ \vdots & & & \vdots \\ 0 & \cdots & \cdots & 0 \end{bmatrix} V^*$$
- we want F so that $\operatorname{rank}(C+F) \le n$ so need to zero out σ_{n+1}, i.e., we want
$$C+F = U \begin{bmatrix} \sigma_1 & & & \\ & \ddots & & \\ & & \sigma_n & \\ & & & 0 \\ 0 & \cdots & \cdots & 0 \\ \vdots & & & \vdots \\ 0 & \cdots & \cdots & 0 \end{bmatrix} V^* \tag{8.3}$$
- so pick
$$F = U \begin{bmatrix} 0 & & & \\ & \ddots & & \\ & & 0 & \\ & & & -\sigma_{n+1} \\ 0 & \cdots & \cdots & 0 \\ \vdots & & & \vdots \\ 0 & \cdots & \cdots & 0 \end{bmatrix} V^*$$
and note that this F would produce the effect needed for (8.3)
- let $V = [\mathbf{v}_1, \dots, \mathbf{v}_{n+1}] \in \mathbb{C}^{(n+1)\times(n+1)}$ where $\mathbf{v}_i \in \mathbb{C}^{n+1}$ is the ith column of V note that $\mathbf{v}_i^*\mathbf{v}_{n+1} = 0$ for all $i = 1, \dots, n$

- we have

$$(C+F)\mathbf{v}_{n+1} = U \begin{bmatrix} \sigma_1 & & & \\ & \ddots & & \\ & & \sigma_n & \\ & & & 0 \\ 0 & \cdots & \cdots & 0 \\ \vdots & & & \vdots \\ 0 & \cdots & \cdots & 0 \end{bmatrix} V^*\mathbf{v}_{n+1} = U \begin{bmatrix} \sigma_1 \mathbf{v}_1^* \\ \vdots \\ \sigma_n \mathbf{v}_n^* \\ \mathbf{0}^\mathsf{T} \\ \mathbf{0}^\mathsf{T} \\ \vdots \\ \mathbf{0}^\mathsf{T} \end{bmatrix} \mathbf{v}_{n+1} = U \begin{bmatrix} \sigma_1 \mathbf{v}_1^* \mathbf{v}_{n+1} \\ \vdots \\ \sigma_n \mathbf{v}_n^* \mathbf{v}_{n+1} \\ \mathbf{0}^\mathsf{T}\mathbf{v}_{n+1} \\ \mathbf{0}^\mathsf{T}\mathbf{v}_{n+1} \\ \vdots \\ \mathbf{0}^\mathsf{T}\mathbf{v}_{n+1} \end{bmatrix} = U \begin{bmatrix} 0 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ 0 \end{bmatrix} = \mathbf{0}$$

- so the vector $\mathbf{v}_{n+1}$ ought to be a candidate for the solution $\mathbf{z}$ in (8.1) but there is one caveat — the last coordinate of $\mathbf{z}$ must be -1 by (8.2)
- how do we achieve that? we divide $\mathbf{v}_{n+1}$ by the negative of its last coordinate, so

$$\begin{bmatrix} \mathbf{x} \\ -1 \end{bmatrix} = \mathbf{z} = -\frac{1}{v_{n+1,n+1}} \mathbf{v}_{n+1}$$

  provided that $v_{n+1,n+1} \neq 0$
- this gives the solution

$$\mathbf{x} = -\begin{bmatrix} v_{1,n+1}/v_{n+1,n+1} \\ \vdots \\ v_{n,n+1}/v_{n+1,n+1} \end{bmatrix}$$

  where the v_{ij} refers to the entries of $V = [v_{ij}]_{i,j=1}^{n+1}$

9. Other applications

- in the homework you see yet other uses of SVD
- here are some other uses of SVD that we didn't have time to consider:
  - least squares with linear constraints (we will discuss this under QR though)
  - least squares with quadratic constraints
  - finding angles between subspaces
  - orthonormal basis for intersection of subspaces
  - subset selection
- all these should convince you that SVD truly is a swiss army knife of matrix computations