DYSCO course on low-rank approximation and its applications

Introduction

Ivan Markovsky

Vrije Universiteit Brussel

Outline

Outline

Subject areas

"You can learn only what you have already half known." *R. Vaccaro*

- numerical linear algebra
  - (generalized) least norm and least squares
  - structured (Hankel/Toeplitz) matrices
  - variable projections method
- optimization
  - penalty methods for nonlinear optimization
  - optimization on a manifold
  - convex relaxations

- statistics
  - errors-in-variables models
  - maximum likelihood
  - bias correction

- dynamical system
  - realization theory, system identification
  - behavioral approach

- computer algebra
  - approximate common divisors
  - polynomial factorizations

- computer vision
  - image deblurring (blind deconvolution)
  - image compression

Aim

"If you try to say everything,
you end up saying nothing." *P. Stewart*

- main goal: recognize and exploit common features, methods, and algorithms across different applications
- low-rank approx. is a unifying problem; related to
  - total least squares (numerical linear algebra)
  - principal component analysis (statistics)
  - factor analysis (psychometrics)
  - latent semantic analysis (natural language proc.)
  - …

Plan

1. Introduction (this lecture)
2. Computational tools (QR, SVD, LS, TLS)
3. Behavioral approach (TLS $\rightarrow$ LRA)
4. System identification (modeling from data)
5. Subspace methods (exact modeling)
6. Generalizations (missing data, . . .)

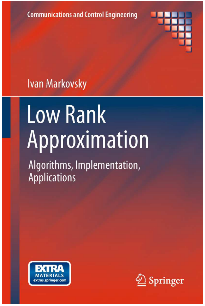
Exercises and evaluation

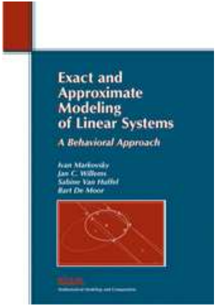
"I hear, I forget;
I see, I remember;
I do, I understand." *Chinese philosopher*

- analytic/computer exercises are part of the course
  - bring a laptop
  - try all problems
- need evaluation? (contact me in the break)
  - work on an individual project, related to the course and feasible to complete in two weeks
  - submit $<$ 10 pages *report* by 21 March and give a 10-minutes *presentation* on 21 March

Materials

- books





- lecture slides available from after the lectures

  http://homepages.vub.ac.be/~imarkovs/dysco

- references to the literature

Outline

"There are repeated patterns in the history of science that teach us how to overcome modern problems. Those who are not aware of the history are missing much." *P. Stewart*

- G. W. Stewart. On the early history of the singular value decomposition. *SIAM Review*, 35(4):551–566, 1993

Eugenio Beltrami (1835–1900)

- considered bilinear forms: $f(x,y) = x^\top A y$
- problem: represent f as a sum of squares via orthogonal transformations $\widetilde{x} = U^\top x$ and $\widetilde{y} = V^\top y$, *i.e.*,

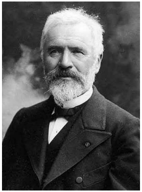
  $f(x,y) = x^\top A y = \widetilde{x}^\top U^\top A V \widetilde{y} = \widetilde{x}^\top \Sigma \widetilde{y}$, with Σ diagonal

- equivalent problem is

  $A = U \Sigma V^\top$, with U, V orthogonal and Σ diagonal

  $\rightsquigarrow$ singular value decomposition $\approx$ low-rank approx.

Camille Jordan (1838–1922)



- problem:

$$\text{maximize} \quad x^\top A y \quad \text{subject to} \quad x^\top x = 1 \text{ and } y^\top y = 1$$

- the solution is given by the extreme singular values and corresponding singular vectors of A

J. Sylvester (1814–1897), E. Schmidt (1876–1959), H. Weyl (1885–1955)



- generalization to infinite dimensional spaces (integrals rather than sums)
- "the fundamental theorem"

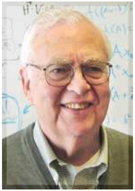
  For any $\widehat{A}$ *with* $\mathrm{rank}(\widehat{A}) \leq r$, $\|A - \widehat{A}\|_2 \geq \sigma_{r+1}(A)$.

Harold Hotelling (1895–1973)

- principal component analysis problem (1933)

  If x is a random vector with zero mean and dispersion D, with eigenvalue decomposition $D = V\Sigma^2 V^\top$, the components of $V^\top x$ are uncorrelated with variances σ_i^2. Then the $\widehat{V}$ factor, obtained from the SVD of X, is an estimate of V.

G. Golub (1932–2007) and W. Kahan (1933–)



- computation of the SVD by a two step procedure:
    1. reduction to a bidiagonal matrix (in $O(mn^2)$ for $m > n$)
    2. compute the SVD of the bidiagonal matrix
       (by a variant of the QR algorithm for EVD)

- step 2 requires iterative algorithms

- convergence to machine precision is fast

- in fact, the first step is more expensive

Rudolf Kalman (1930–) and others

- realization theory

$$\text{rank(Hankel matrix)} = \text{order of a minimal realization}$$

- B. Moore. Principal component analysis in linear systems: Controllability, observability and model reduction. *IEEE Trans. Automat. Control*, 26(1):17–31, 1981

- nuclear norm heuristic for rank minimization problems

Outline

"For applicable control engineering research, three things need to be present:

1. *a real and pressing set of problems,*
2. *intuitively graspable theoretical approaches to design, which can be underpinned by sound mathematics, and*
3. *good interactive software which can be used to turn designs into practical applications."*

A. MacFarlane

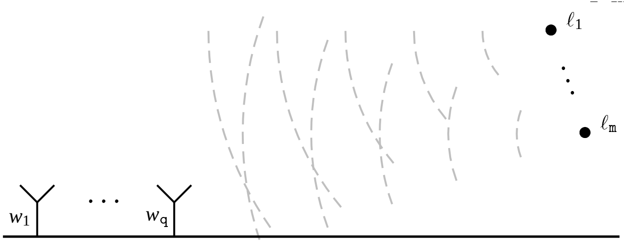
- A. MacFarlane. Multivariable feedback: a personal reminiscence. *International Journal of Control*, 86(11):1903–1923, 2013
- next set of problems (Section 1.3 of the book)

Applications

1. Direction of arrival estimation (signal processing)
2. Latent semantic analysis (language processing)
3. Recommender systems (machine learning)
4. Multidimensional scaling (computer vision)
5. Conic section fitting (computer vision)
6. System realization (systems and control)
7. System identification (systems and control)
8. Greatest common divisor (computer algebra)

Direction of arrival estimation

- setup: $\mathtt{q}$ antennas and $\mathtt{m} < \mathtt{q}$ distant sources



- ℓ_k — source intensity (a function of time)
- $w(t) = p_k \ell_k(t - \tau_k)$ — array's response to kth source
- τ_k — pure delay

- p_k depends on the array geometry and the source locations (assumed constant in time)

- assuming that the array responds linearly to a mixture of sources, we have

$$
\begin{aligned}
D &= \begin{bmatrix} w(1) & \cdots & w(T) \end{bmatrix} \\
&= \sum_{k=1}^{\mathrm{m}} p_k \underbrace{\begin{bmatrix} \ell_k(1-\tau_k) & \cdots & \ell_k(T-\tau_k) \end{bmatrix}}_{\ell_k} = PL
\end{aligned}
$$

where $P := \begin{bmatrix} p_1 & \cdots & p_{\mathrm{m}} \end{bmatrix}$ and $L := \begin{bmatrix} \ell_1 \\ \vdots \\ \ell_{\mathrm{m}} \end{bmatrix}$

- $\operatorname{rank}(D) = \#$ of sources

Computational problem

- with exact data D, the direction of arrival problem is

  rank revealing factorization PL of D

- P, L carry information about the source locations

- in practice, D is full rank and we aim to

  approximate D by $\widehat{D}$ *of rank* $\leq \mathtt{m} < \max(q, N)$

- this is

  unstructured low-rank approximation problem

Notes

- the rank constraint $\mathtt{m}$ is a hyper parameter
- determining its value is part of the problem
- from $\widehat{D}$, we need to obtain P, L, such that $\widehat{D} = PL$
- this is the (simple) problem of exact modeling (rank-revealing factorization)
- some algorithms return P, L as a byproduct
- we separate the issues of
  1. solution methods (optimization algorithms)
  2. problem formulation (low-rank approximation)

Latent semantic analysis

- N documents involve q terms and $\mathtt{m}$ concepts
- p_k — term frequencies related to the kth concept
- ℓ_{kj} — relevance of the kth concept to the jth document
- the term frequencies related to the documents are

$$D = \sum_{k=1}^{\mathtt{m}} p_k \underbrace{\begin{bmatrix} \ell_{k1} & \cdots & \ell_{kN} \end{bmatrix}}_{\ell_k} = \begin{bmatrix} p_1 & \cdots & p_{\mathtt{m}} \end{bmatrix} \begin{bmatrix} \ell_1 \\ \vdots \\ \ell_{\mathtt{m}} \end{bmatrix} = PL$$

- $\mathrm{rank}(D) = \#$ of concepts

Recommender systems

- q items are rated by N users
- d_{ij} — rating of the ith item by the jth user
- not all ratings are available $\rightsquigarrow$ missing data in D
- assumption: $\mathtt{m}$ "typical" users, where $\mathtt{m} \ll \min(q, N)$
- p_k — ratings of the items by the kth typical user

- the *j*th user is a linear combination of typical users

$$d_j = \sum_{k=1}^{\mathtt{m}} p_k \ell_{kj}$$

$\ell_k := \begin{bmatrix} \ell_{k1} & \cdots & \ell_{kN} \end{bmatrix}$ — weights for the *j*th user

- model for the ratings

$$D = \sum_{k=1}^{\mathtt{m}} p_k \ell_k = PL$$

- $\mathrm{rank}(D) =$ number of “typical” users

Matrix completion problems

- exact matrix completion

$$\begin{array}{lll} \text{minimize} & \text{over } \widehat{D} & \operatorname{rank}(\widehat{D}) \\ \text{subject to} & \widehat{D}_{ij} = D_{ij} & \text{for all } (i,j) \text{, where } D_{ij} \text{ is given} \end{array}$$

- approximate matrix completion

$$\begin{array}{lll} \text{minimize} & \text{over } \widehat{D} \text{ and } \Delta D & \operatorname{rank}(\widehat{D}) + \lambda \|\Delta D\|_{\mathrm{F}} \\ \text{subject to} & \widehat{D}_{ij} = D_{ij} + \Delta D_{ij} & \text{for all } (i,j) \text{, where } D_{ij} \text{ is given} \end{array}$$

Multidimensional scaling

- consider N points: $\mathscr{X} := \{ x_1, \ldots, x_N \} \subset \mathbb{R}^2$
- $d_{ij} := \|x_i - x_j\|_2^2$ — squared distance from x_i to x_j
- distance matrix: $D = \left[d_{ij} \right]$ of the pair-wise distances
- $\operatorname{rank}(D) \leq 4$, indeed

$$d_{ij} = (x_i - x_j)^\top (x_i - x_j) = x_i^\top x_i - 2 x_i^\top x_j + x_j^\top x_j$$

$$d_{ij} = (x_i - x_j)^\top (x_i - x_j) = x_i^\top x_i - 2x_i^\top x_j + x_j^\top x_j$$

$$D = \underbrace{\begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \begin{bmatrix} x_1^\top x_1 & \cdots & x_N^\top x_N \end{bmatrix}}_{\text{rank } \leq 1} - 2 \underbrace{\begin{bmatrix} x_1^\top \\ \vdots \\ x_N^\top \end{bmatrix} \begin{bmatrix} x_1 & \cdots & x_N \end{bmatrix}}_{\text{rank } \leq 2}$$

$$+ \underbrace{\begin{bmatrix} x_1^\top x_1 \\ \vdots \\ x_N^\top x_N \end{bmatrix} \begin{bmatrix} 1 & \cdots & 1 \end{bmatrix}}_{\text{rank } \leq 1}$$

- approximate modeling:

  bilinearly structured low-rank approximation

Conic section fitting

- data:

$$\{ d_1, \dots, d_N \} \subset \mathbb{R}^2, \qquad \text{where} \quad d_j = \begin{bmatrix} a_j \\ b_j \end{bmatrix}$$

- model:

$$\mathscr{B}(S,u,v) := \{ d \in \mathbb{R}^2 \mid d^\top S d + u^\top d + v = 0 \}$$

- linear relation in the model parameters

$$d^\top S d + u^\top d + v = \begin{bmatrix} s_{11} & 2s_{12} & u_1 & s_{22} & u_2 & v \end{bmatrix} \begin{bmatrix} a^2 \\ ab \\ a \\ b^2 \\ b \\ 1 \end{bmatrix}$$

- parameter vector

$$\theta := \begin{bmatrix} s_{11} & 2s_{12} & u_1 & s_{22} & u_2 & v \end{bmatrix}$$

- extended data vector (feature map)

$$d_{\text{ext}} := \begin{bmatrix} a^2 & ab & a & b^2 & b & 1 \end{bmatrix}^\top$$

- exact modeling

$$d \in \mathscr{B}(\theta) = \mathscr{B}(S, u, v) \qquad \Longleftrightarrow \qquad \theta d_{\text{ext}} = 0$$

- approximate modeling:

  quadratically structured low-rank approximation

System realization

- problem:

  impulse response $\mapsto$ *state space representation*

- let H be an impulse response of $\mathtt{n}$th order discrete-time linear time-invariant system

- then

$$\operatorname{rank} \underbrace{\begin{bmatrix} H(1) & H(2) & H(3) & \cdots \\ H(2) & H(3) & \iddots & \\ H(3) & \iddots & & \\ \vdots & & & \end{bmatrix}}_{\mathscr{H}(H)} = \mathtt{n}$$

- partial realization problem

Stochastic realization

- stochastic system: $\text{white noise} \longrightarrow \boxed{\text{deterministic system}} \longrightarrow y$

- data: $R(\tau) := \mathbf{E}\big(y(t)y^\top(t-\tau)\big)$ autocorrelation

- problem:

  autocorrelation $R \mapsto$ state space representation

- main result:

$$\operatorname{rank}\big(\mathscr{H}(R)\big) = \text{order of minimal realization of } R$$

System identification

- problem:

  general trajectory $\mapsto$ *representation of the system*

- data:

$$w = \begin{bmatrix} u \\ y \end{bmatrix}, \qquad \begin{aligned} u &= \big(u(1), \dots, u(T)\big) \text{ — input} \\ y &= \big(y(1), \dots, y(T)\big) \text{ — output} \end{aligned}$$

- link to low-rank approximation

$$\operatorname{rank}\big(\mathscr{H}_{\mathrm{n}_{\max}+1}(w)\big) \leq \operatorname{rank}\big(\mathscr{H}_{\mathrm{n}_{\max}+1}(u)\big) + \text{order of the system}$$

- persistency of excitation: $\mathscr{H}(u)$ is full row rank

Greatest common divisor

- the GCD of the polynomials

$$p(z) = p_0 + p_1 z + \cdots + p_n z^n$$
$$q(z) = q_0 + q_1 z + \cdots + q_m z^m$$

is polynomial c of maximal degree dividing p and q

$$p = rc \qquad \text{and} \qquad q = sc$$

- main result:

$$\text{degree}(c) = n + m - \text{rank}\left(\mathscr{S}(p,q)\right)$$

$\mathscr{S}(p,q)$ — $(n+m) \times (n+m)$ Sylvester matrix

The Sylvester matrix of p and q

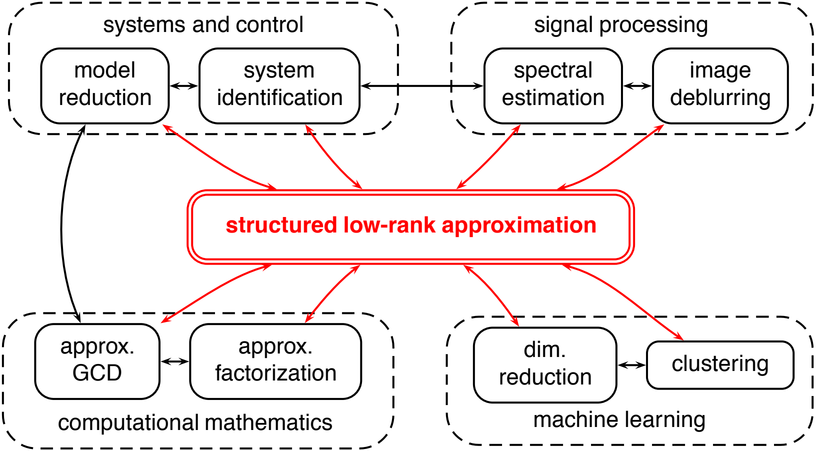
$$\mathscr{S}(p,q) := \begin{bmatrix} p_0 & & & & q_0 & & & \\ p_1 & p_0 & & & q_1 & q_0 & & \\ \vdots & p_1 & \ddots & & \vdots & q_1 & \ddots & \\ p_n & \vdots & \ddots & p_0 & q_m & \vdots & \ddots & q_0 \\ & p_n & & p_1 & & q_m & & q_1 \\ & & \ddots & \vdots & & & \ddots & \vdots \\ & & & p_n & & & & q_m \end{bmatrix}$$

an $(m+n) \times (m+n)$ structured matrix

Other applications

- Factor analysis (psychometrics)
- Multivariate calibration (chemometrics)
- Microarray data analysis (bioinformatics)
- Fundamental matrix estimation (computer vision)
- Factorizability of multivariable polynomials

One problem, many applications



Outline

IQ test

- extend the sequence: 0, 1, 1, 2, 3, 5, 8, ...
- extend the sequence: 0, 1, 1, 2, 5, 9, 18, ...
- more interesting is to find a systematic solution
- the key ingredient is rank deficiency of a matrix

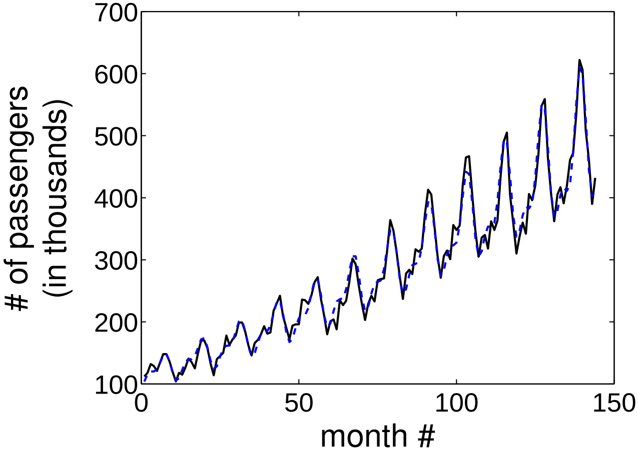
  "Behind every data modeling problem there is a (hidden) low-rank approximation problem: the model imposes relations on the data which render a matrix constructed from exact data rank deficient."

Time series interpolation

- from extrapolation to interpolation
- data: classic Box & Jenkins airline data
  monthly airline passenger numbers 1949–1960
- aim: estimate missing values
  - missing values in "the future": extrapolation
  - other missing values: interpolation
  - take into account the time series nature of the data

Autonomous LTI model

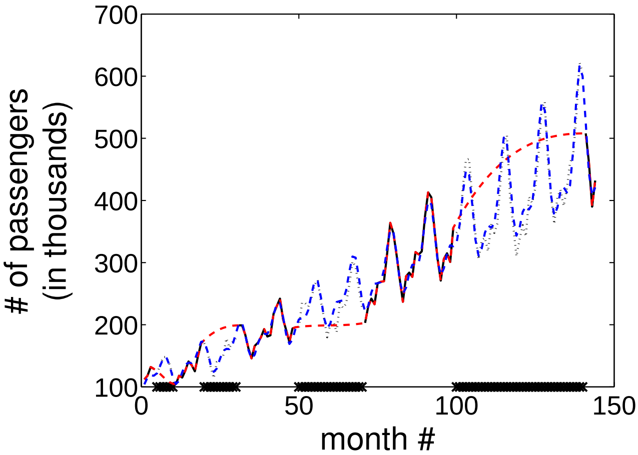
- using all 144 data points to identify a model



- solid line — data, dashed — fit by 6th order model

Missing data estimation

- `[5:10 20:30 50:70 100:140]` are missing



- piecewise cubic interpolation, 6th order LTI model

Modeling as data compression

- the model is a concise representation of the data
- exact model $\leftrightarrow$ lossless compression (*e.g.*, `zip`)
- approximate model $\leftrightarrow$ lossy compression (*e.g.*, `mp3`)

Example: compression of a random vector

- data: $1 \times n$ vector, generated by `randn`
- compression in `mat` format

	length n	1	223	334	556	667	1000
1.	original size	8	1784	2672	4448	5336	8000
2.	mat file size	178	1945	2798	4490	5341	7893

Example: low-rank matrix compression

- data: random 100×100 matrix D of rank 5

- stored in four different ways

	representation	size
1.	all elements of D	80000
2.	D in `mat` format	75882
3.	all elements of P and L	8024
4.	P and L in `mat` format	7767

- in 2 and 4, we compute a rank revealing factorization

$$D = PL$$

- can we do better than storing P and L (compressed)?

Example: trajectory of an LTI system

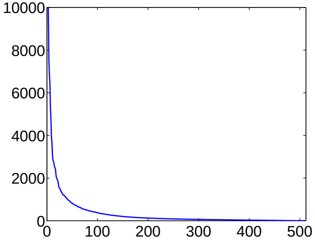
- data: impulse response of a random 3rd order system
- stored in four different ways

	representation	size
1.	impulse response h	192
2.	h in `mat` format	377
3.	model parameters θ	56
4.	θ in `mat` format	233

- in 3 and 4, we have parameterized the system

Low-rank approximation of images

- an image is a matrix of gray values (integers 0–255)
- typical singular values plot:



- $\Longrightarrow$ an image can be approximate by lower rank
- the basis of many methods for image processing
- note that SVD does not respect the 0–255 bounds

Original 512×512 image

Rank 100 approximation

Rank 80 approximation

Rank 60 approximation

Rank 40 approximation

Outline

About the course

Historical review

Applications

Demos

Exercises