Chapter 1
Introduction

The very art of mathematics is to say the same thing another way.

Unknown

1.1 Classical and behavioural paradigms for data modeling

Fitting linear models to data can be achieved, both conceptually and algorithmically, by solving approximately a linear system of equations

$$AX \approx B, \tag{LSE}$$

where the matrices A and B are constructed from the given data and the matrix X parametrizes the model. In this *classical paradigm*, the main tools are the ordinary linear least squares method and its variations—regularized least squares, total least squares, robust least squares, *etc*. The least squares method and its variations are mainly motivated by their applications for data fitting, but they invariably consider solving approximately an overdetermined system of equations.

The underlying premise in the classical paradigm is that existence of an exact linear model for the data is equivalent to existence of solution X to a system $AX = B$. Such a model is a linear map: the variables corresponding to the A matrix are inputs (or causes) and the variables corresponding to the B matrix are outputs (or consequences) in the sense that they are determined by the inputs and the model. Note that in the classical paradigm the input/output partition of the variables is postulated a priori. Unless the model is required to have the a prior specified input/output partition, imposing such structure in advance is ad hoc and leads to undesirable theoretical and numerical features of the modeling methods derived.

An alternative to the classical paradigm that does not impose an a priori fixed input/output partition is the *behavioral paradigm*. In the behavioral paradigm, fitting linear models to data is equivalent to the problem of approximating a matrix D, constructed from the data, by a matrix $\widehat{D}$ of lower rank. Indeed, existence of an exact linear model for D is equivalent to D being rank deficient. Moreover, the rank of D is related to the complexity of the model. This fact is the tenet of the book and is revisited in the following chapters in the context of applications from systems and control, signal processing, computer algebra, and machine learning. Also its implication to the development of numerical algorithms for data fitting is explored.

To see that existence of a low complexity exact linear model is equivalent to rank deficiency of the data matrix, let the columns $d_1, \dots, d_N$ of D be the observations and the elements $d_{1j}, \dots, d_{\mathtt{q}j}$ of d_j be the observed variables. We assume that there are at least as many observations as observed variables, *i.e.*, $\mathtt{q} \leq N$. A linear model for D declares that there are linear relations among the variables, *i.e.*, there are vectors r_k, such that

$$r_k^\top d_j = 0, \qquad \text{for } j = 1, \dots, N.$$

If there are $\mathtt{p}$ independent linear relations, then D has rank less than or equal to $\mathtt{m} := \mathtt{q} - \mathtt{p}$ and the observations belong to at most $\mathtt{m}$-dimensional subspace $\mathscr{B}$ of $\mathbb{R}^{\mathtt{q}}$. We identify the model for D, defined by the linear relations $r_1, \dots, r_{\mathtt{p}} \in \mathbb{R}^{\mathtt{q}}$, with the set $\mathscr{B} \subset \mathbb{R}^{\mathtt{q}}$. Once a model $\mathscr{B}$ is obtained from the data, all possible input/output partitions can be enumerated, which is an analysis problem for the identified model. Therefore, the choice of an input/output partition in the behavioral paradigm to data modeling can be incorporated, if desired, in the modeling problem and thus need not be hypothesized as necessarily done in the classical paradigm.

The classical and behavioral paradigms for data modeling are related but not equivalent. Although existence of solution of the system $AX = B$ implies that the matrix $\begin{bmatrix} A & B \end{bmatrix}$ is low rank, it is not true that $\begin{bmatrix} A & B \end{bmatrix}$ having a sufficiently low rank implies that the system $AX = B$ is solvable. This lack of equivalence causes ill-posed (or numerically ill-conditioned) data fitting problems in the classical paradigm, which have no solution (or are numerically difficult to solve). In terms of the data fitting problem, ill-conditioning of the problem (LSE) means that the a priori fixed input/output partition of the variables is not corroborated by the data. In the behavioral setting without the a priori fixed input/output partition of the variables, ill-conditioning of the data matrix D implies that the data approximately satisfies linear relations, so that ill-conditioning is a good feature of the data.

The classical paradigm is included in the behavioral paradigm as a special case because approximate solution of an overdetermined system of equations (LSE) is a possible approach to achieve low rank approximation. Alternatively, low rank approximation can be achieved by approximating the data matrix with a matrix that has at least $\mathtt{p}$-dimensional null space, or at most $\mathtt{m}$-dimensional column space. Parametrizing the null space and the column space by sets of basis vectors, the alternative approaches are:

1. *kernel representation* there is a full row rank matrix $R \in \mathbb{R}^{\mathtt{p} \times \mathtt{q}}$, such that

$$RD = 0,$$

2. *image representation* there are matrices $P \in \mathbb{R}^{\mathtt{q} \times \mathtt{m}}$ and $L \in \mathbb{R}^{\mathtt{m} \times N}$, such that

$$D = PL.$$

The approaches using kernel and image representations are equivalent to the original low rank approximation problem. Next, the use of $AX = B$, kernel, and image representations is illustrated on the most simple data fitting problem—line fitting.

1.2 Motivating example for low rank approximation

Given a (multi)set of points $\{d_1,\ldots,d_N\} \subset \mathbb{R}^2$ in the plane, the aim of the line fitting problem is to find a line passing through the origin that "best" matches the given points. The classical approach for line fitting is to define

$$\begin{bmatrix} a_j \\ b_j \end{bmatrix} = \operatorname{col}(a_j,b_j) := d_j$$

and solve approximately the overdetermined system

$$\operatorname{col}(a_1,\ldots,a_N)x = \operatorname{col}(b_1,\ldots,b_N) \tag{lse}$$

by the least squares method. Let x_{ls} be the least squares solution to (lse). Then the least squares fitting line is

$$\mathscr{B}_{\text{ls}} := \{\, d = \operatorname{col}(a,b) \in \mathbb{R}^2 \mid ax_{\text{ls}} = b \,\}.$$

Geometrically, $\mathscr{B}_{\text{ls}}$ minimizes the sum of the squared *vertical distances* from the data points to the fitting line.

The left plot in Figure 1.1 shows a particular example with $N = 10$ data points. The data points $d_1,\ldots,d_{10}$ are the circles in the figure, the fit $\mathscr{B}_{\text{ls}}$ is the solid line, and the fitting errors $e := ax_{\text{ls}} - b$ are the dashed lines. Visually one expects the best fit to be the vertical axis, so minimizing vertical distances does not seem appropriate in this example.

Note that by solving (lse), a (the first components of the d) is treated differently from b (the second components): b is assumed to be a *function* of a. This is an arbitrary choice; the data can be fitted also by solving approximately the system

$$\operatorname{col}(a_1,\ldots,a_N) = \operatorname{col}(b_1,\ldots,b_N)x, \tag{lse'}$$

in which case a is assumed to be a function of b. Let x'_{ls} be the least squares solution to (lse'). It gives the fitting line

$$\mathscr{B}'_{\text{ls}} := \{\, d = \operatorname{col}(a,b) \in \mathbb{R}^2 \mid a = bx'_{\text{ls}} \,\},$$

which minimizes the sum of the squared *horizontal distances* (see the right plot in Figure 1.1). The line $\mathscr{B}'_{\text{ls}}$ happens to achieve the desired fit in the example.

In the classical approach for data fitting, *i.e.*, solving approximately a linear system of equations in the least squares sense, the choice of the model representation affects the fitting criterion.

This feature of the classical approach is undesirable: it is more natural to specify a desired fitting criterion independently of how the model happens to be parametrized.

In many data modeling methods, however, a model representation is a priori fixed and implicitly corresponds to a particular fitting criterion.

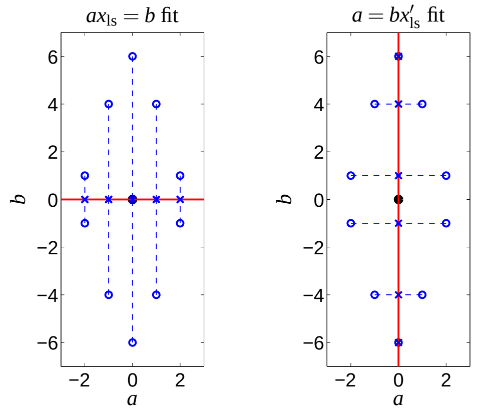

Fig. 1.1 Least squares fits (solid lines) minimizing vertical (left plot) and horizontal (right plot) distances.

The total least squares method is an alternative to least squares method for solving approximately an overdetermined system of linear equations. In terms of data fitting, the total least squares method minimizes the sum of the squared *orthogonal distances* from the data points to the fitting line. Using the system of equations (lse), line fitting by the total least squares method leads to the problem

$$\begin{aligned} &\text{minimize} \quad \text{over } x \in \mathbb{R},\ \begin{bmatrix} \widehat{a}_1 \\ \vdots \\ \widehat{a}_N \end{bmatrix} \in \mathbb{R}^N, \text{ and } \begin{bmatrix} \widehat{b}_1 \\ \vdots \\ \widehat{b}_N \end{bmatrix} \in \mathbb{R}^N \quad \sum_{j=1}^{N} \left\| d_j - \begin{bmatrix} \widehat{a}_j \\ \widehat{b}_j \end{bmatrix} \right\|_2^2 \\ &\text{subject to} \quad \widehat{a}_j x = \widehat{b}_j, \quad \text{for } j = 1,\ldots,N. \end{aligned} \tag{tls}$$

However, for the data in Figure 1.1 the total least squares problem has no solution. Informally, the approximate solution is $x_{\text{tls}} = \infty$, which corresponds to a fit by a vertical line. Formally,

the total least squares problem (tls) may have no solution and therefore fail to give a model.

The use of (lse) in the definition of the total least squares line fitting problem restricts the fitting line to be a graph of a function $ax = b$ for some $x \in \mathbb{R}$. Thus, the vertical line is a priori excluded as a possible solution. In the example, the line minimizing the sum of the squared orthogonal distances happens to be the vertical line. For this reason, x_{tls} does not exist.

Any line $\mathscr{B}$ passing through the origin can be represented as an image and a kernel, *i.e.*, there exist matrices $P \in \mathbb{R}^{2\times 1}$ and $R \in \mathbb{R}^{1\times 2}$, such that

$$\mathscr{B} = \operatorname{image}(P) := \{\, d = P\ell \in \mathbb{R}^2 \mid \ell \in \mathbb{R} \,\}$$

and

$$\mathscr{B} = \ker(R) := \{\, d \in \mathbb{R}^2 \mid Rd = 0 \,\}.$$

Using the image representation of the model, the line fitting problem of minimizing the sum of the squared orthogonal distances is

$$\begin{aligned} &\text{minimize} && \text{over } P \in \mathbb{R}^{2\times 1} \text{ and } \begin{bmatrix}\ell_1 & \cdots & \ell_N\end{bmatrix} \in \mathbb{R}^{1\times N} \quad \sum_{j=1}^{N} \|d_j - \widehat{d}_j\|_2^2 \\ &\text{subject to} && \widehat{d}_j = P\ell_j, \quad \text{for } j = 1,\dots,N. \end{aligned} \tag{lra$'_P$}$$

With

$$D := \begin{bmatrix} d_1 & \cdots & d_N \end{bmatrix}, \qquad \widehat{D} := \begin{bmatrix} \widehat{d}_1 & \cdots & \widehat{d}_N \end{bmatrix},$$

and $\|\cdot\|_{\mathrm{F}}$ the *Frobenius norm*,

$$\|E\|_{\mathrm{F}} := \big\|\operatorname{vec}(E)\big\|_2 = \Big\| \begin{bmatrix} e_{11} & \cdots & e_{\mathtt{q}1} & \cdots & e_{1N} & \cdots & e_{\mathtt{q}N} \end{bmatrix}^\top \Big\|_2, \quad \text{for all } E \in \mathbb{R}^{\mathtt{q}\times N}$$

(lra$'_P$) is more compactly written as

$$\begin{aligned} &\text{minimize} && \text{over } P \in \mathbb{R}^{2\times 1} \text{ and } L \in \mathbb{R}^{1\times N} \quad \|D - \widehat{D}\|_{\mathrm{F}}^2 \\ &\text{subject to} && \widehat{D} = PL. \end{aligned} \tag{lra$_P$}$$

Similarly, using a kernel representation, the line fitting problem, minimizing the sum of squares of the orthogonal distances is

$$\begin{aligned} &\text{minimize} && \text{over } R \in \mathbb{R}^{1\times 2},\ R \neq 0, \text{ and } \widehat{D} \in \mathbb{R}^{2\times N} \quad \|D - \widehat{D}\|_{\mathrm{F}}^2 \\ &\text{subject to} && R\widehat{D} = 0. \end{aligned} \tag{lra$_R$}$$

Contrary to the total least squares problem (tls), problems (lra$_P$) and (lra$_R$) always have (nonunique) solutions. In the example, solutions are, *e.g.*, $P^* = \operatorname{col}(0,1)$ and $R^* = \begin{bmatrix} 1 & 0 \end{bmatrix}$, which describe the vertical line

$$\mathscr{B}^* := \operatorname{image}(P^*) = \ker(R^*).$$

The constraints

$$\widehat{D} = PL, \text{ with } P \in \mathbb{R}^{2\times 1},\ L \in \mathbb{R}^{1\times N} \quad \text{and} \quad R\widehat{D} = 0, \text{ with } R \in \mathbb{R}^{1\times 2},\ R \neq 0$$

are equivalent to the constraint $\operatorname{rank}(\widehat{D}) \leq 1$, which shows that the points $\{\, \widehat{d}_1, \dots, \widehat{d}_N \,\}$ being fitted exactly by a line passing through the origin is equivalent to

$$\operatorname{rank}\big(\begin{bmatrix} \widehat{d}_1 & \cdots & \widehat{d}_N \end{bmatrix}\big) \leq 1.$$

Thus, (lra$_P$) and (lra$_R$) are instances of one and the same

abstract problem: approximate the data matrix D by a low rank matrix $\widehat{D}$.

In Chapter 2, we generalize the observations made in the line fitting example to modeling of $\mathtt{q}$-dimensional data. The underlying goal is:

Given a set of points in $\mathbb{R}^{\mathtt{q}}$ (the data), find a subspace of $\mathbb{R}^{\mathtt{q}}$ of bounded dimension (a model) that has the least (2-norm) distance to the data points.

Such a subspace is a (2-norm) optimal fitting *model*. General *representations* of a subspace in $\mathbb{R}^{\mathtt{q}}$ are the kernel or the image of a matrix. The classical least squares and total least squares formulations of the data modeling problem exclude some subspaces. The equations $AX = B$ and $A = BX'$, used in the least squares and total least squares problem formulations to represent the subspace, might fail to represent the optimal solution, while the kernel and image representations do not have such deficiency. This suggests that the kernel and image representations are better suited for data modeling.

The equations $AX = B$ and $A = BX'$ were introduced from an algorithmic point of view—by using them, the data fitting problem is turned into the standard problem of solving approximately an overdetermined linear system of equations. An interpretation of these equations in the data modeling context is that in the model represented by the equation $AX = B$, the variable A is an input and the variable B is an output. Similarly, in the model represented by the equation $A = BX$, A is an output and B is an input. The input/output interpretation has an intuitive appeal because it implies a causal dependence of the variables: the input is causing the output.

Representing the model by an equation $AX = B$ and $A = BX'$, as done in the classical approach, one a priori assumes that the optimal fitting model has a certain input/output structure. The consequences are:

- existence of exceptional (nongeneric) cases, which complicate the theory,
- ill-conditioning caused by "nearly" exceptional cases, which leads to lack of numerical robustness of the algorithms, and
- need of regularization, which leads to a change of the specified fitting criterion.

These aspects of the classical approach are generally considered as inherent to the data modeling problem. By choosing the alternative image and kernel model representations, the problem of solving approximately an overdetermined system of equations becomes a low rank approximation problem, where the nongeneric cases (and the related issues of ill-conditioning and need of regularization) are avoided.

1.3 Overview of applications

In this section, we list examples of low rank approximation drawn from different application areas. The fact that a matrix constructed from exact data is low rank and the approximate modeling problem is low rank approximation is sometimes well known (*e.g.*, in realization theory, model reduction, and approximate greatest common divisor). In other cases (*e.g.*, natural language processing and conic section fitting), the link to low rank approximation is less well known and is not exploited.

Common pattern in data modeling

The motto of the book is:

> Behind every data modeling problems 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.

Although an exact data matrix is low rank, a matrix constructed from observed data is generically full rank due to measurement noise, unaccounted effects, and assumptions about the data generating system that are not satisfied in practice. Therefore, generically, the observed data does not have an exact low complexity model. This leads to the problem of approximate modeling, which can be formulated as a low rank approximation problem as follows. Modify the data as little as possible, so that the matrix constructed from the modified data has a specified low rank. The modified data matrix being low rank implies that there is an exact model for the modified data. This model is by definition an approximate model for the given data. The transition from exact to approximate modeling is an important step in building a coherent theory for data modeling and is emphasized in this book.

In all applications, we first discuss the exact modeling problem and then the practically more important approximate modeling problem. This is done because 1) exact modeling is simpler than approximate modeling, so that pedagogically it is the right starting place, and 2) exact modeling is a part of optimal approximate modeling and suggests ways of solving such problems suboptimally. Indeed, small modifications of exact modeling algorithms lead to effective approximate modeling algorithms. Well known examples of the transition from exact to approximate modeling in systems theory are the progressions from realization theory to model reduction and from deterministic subspace identification to approximate and stochastic subspace identification methods.

The estimator consistency question in stochastic estimation problems corresponds to exact data modeling because asymptotically the true data generating system is recovered from observed data. Estimation with finite sample size, however, necessarily involves approximation. Thus in stochastic estimation theory there is also a step of transition from exact to approximate.

$$\begin{tikzcd}
\text{exact deterministic} \arrow[r] \arrow[d] & \text{approximate deterministic} \arrow[d] \\
\text{exact stochastic} \arrow[r] & \text{approximate stochastic}
\end{tikzcd}$$

Fig. 1.2 Transitions among exact deterministic, approximate deterministic, exact stochastic, and approximate stochastic modeling problems. The arrows show progression from simpler to more complicated.

Applications in systems and control

Deterministic system realization and model reduction

Realization theory addresses the problem of finding a state representation of a linear time-invariant dynamical system defined by a transfer function or impulse response representation. The key result in realization theory is that a sequence

$$H = \big(H(0), H(1), \ldots, H(t), \ldots\big)$$

is an impulse response of a discrete-time linear time-invariant system of order $\mathtt{n}$ if and only if the two sided infinite Hankel matrix

$$\mathscr{H}(H) := \begin{bmatrix} H(1) & H(2) & H(3) & \cdots \\ H(2) & H(3) & ⋰ & \\ H(3) & ⋰ & & \\ \vdots & & & \end{bmatrix},$$

constructed from H has rank $\mathtt{n}$, *i.e.*,

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

Therefore, existence of a finite dimensional realization of H (exact low complexity linear time-invariant model for H) is equivalent to rank deficiency of a Hankel matrix constructed from the data. A minimal state representation can be obtained from a rank revealing factorization of $\mathscr{H}(H)$.

When there is no exact finite dimensional realization of the data or the exact realization is of high order, one may want to find an approximate realization of a specified low order $\mathtt{n}$. These, respectively, approximate realization and model reduction problems naturally lead to *Hankel structured low rank approximation.*

The deterministic system realization and model reduction problems are further considered in Sections 2.2, 3.1, and 4.2.

Stochastic system realization

Let y be the output of an nth order linear time-invariant system, driven by white noise (a stochastic system) and let $\mathbf{E}$ be the expectation operator. The sequence

$$R = \big(R(0), R(1), \dots, R(t), \dots\big)$$

defined by

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

is called the *autocorrelation* sequence of y. Stochastic realization theory is concerned with the problem of finding a state representation of a stochastic system that could have generated the observed output y, *i.e.*, a linear time-invariant system driven by white noise, which output correlation sequence is equal to R.

An important result in stochastic realization theory is that R is the output correlation sequence of an $\mathtt{n}$th order stochastic system if and only if the Hankel matrix $\mathscr{H}(R)$ constructed from R has rank $\mathtt{n}$, *i.e.*,

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

Therefore, stochastic realization of a random process y is equivalent to deterministic realization of its autocorrelation sequence R. When it exists, the finite dimensional stochastic realizations can be obtained from a rank revealing factorization of the matrix $\mathscr{H}(R)$.

In practice, only a finite number of finite length realizations of the output y are available, so that the autocorrelation sequence is *estimated* from y. With an estimate $\widehat{R}$ of the autocorrelation R, the Hankel matrix $\mathscr{H}(\widehat{R})$ is almost certainly full rank, which implies that a finite dimensional stochastic realization can not be found. Therefore, the problem of finding an approximate stochastic realization occurs. This problem is again Hankel structured low rank approximation.

System identification

Realization theory considers a *system representation problem*: pass from one representation of a system to another. Alternatively, it can be viewed as a special exact identification problem: find from impulse response data (a special trajectory of the system) a state space representation of the data generating system. The exact identification problem (also called deterministic identification problem) is to find from a general response of a system, a representation of that system. Let

$$w = \operatorname{col}(u, y), \qquad \text{where} \quad u = \big(u(1), \dots, u(T)\big) \quad \text{and} \quad y = \big(y(1), \dots, y(T)\big)$$

be an input/output trajectory of a discrete-time linear time-invariant system of order $\mathtt{n}$ with $\mathtt{m}$ inputs and $\mathtt{p}$ outputs and let $\mathtt{n}_{\max}$ be a given upper bound on $\mathtt{n}$. Then the Hankel matrix

$$\mathscr{H}_{\mathtt{n}_{\max}+1}(w) := \begin{bmatrix} w(1) & w(2) & \cdots & w(T-\mathtt{n}_{\max}) \\ w(2) & w(3) & \cdots & w(T-\mathtt{n}_{\max}+1) \\ \vdots & \vdots & & \vdots \\ w(\mathtt{n}_{\max}+1) & w(\mathtt{n}_{\max}+1) & \cdots & w(T) \end{bmatrix} \qquad (\mathscr{H}_i)$$

with $\mathtt{n}_{\max}+1$ block rows, constructed from the trajectory w, is rank deficient:

$$\operatorname{rank}\big(\mathscr{H}_{\mathtt{n}_{\max}+1}(w)\big) \le \operatorname{rank}\big(\mathscr{H}_{\mathtt{n}_{\max}+1}(u)\big) + \text{order of the system.} \qquad \text{(SYSID)}$$

Conversely, if the Hankel matrix $\mathscr{H}_{\mathtt{n}_{\max}+1}(w)$ has rank $(\mathtt{n}_{\max}+1)\mathtt{m}+\mathtt{n}$ and the matrix $\mathscr{H}_{2\mathtt{n}_{\max}+1}(u)$ is full row rank (*persistency of excitation* of u), then w is a trajectory of a controllable linear time-invariant system of order $\mathtt{n}$. Under the above assumptions, the data generating system can be identified from a rank revealing factorization of the matrix $\mathscr{H}_{\mathtt{n}_{\max}+1}(w)$.

When there are measurement errors or the data generating system is not a low complexity linear time-invariant system, the data matrix $\mathscr{H}_{\mathtt{n}_{\max}+1}(w)$ is generically full rank. In such cases, an approximate low-complexity linear time-invariant model for w can be derived by finding a Hankel structured low rank approximation of $\mathscr{H}_{\mathtt{n}_{\max}+1}(w)$. Therefore, the Hankel structured low rank approximation problem can be applied also for approximate system identification.

Similarly, to the analogy between deterministic and stochastic system realization, there is an analogy between deterministic and stochastic system identification. The latter analogy suggests an application of Hankel structured low rank approximation to stochastic system identification.

Linear time-invariant system identification is a major topic in the book and will reappear frequently, see, *e.g.*, Sections 4.3.

Applications in computer algebra

Greatest common divisor of two polynomials

The greatest common divisor of the polynomials

$$p(z) = p_0 + p_1 z + \cdots + p_n z^n \qquad \text{and} \qquad q(z) = q_0 + q_1 z + \cdots + q_n z^m$$

is a polynomial c of maximal degree that divides both p and q, *i.e.*, a maximal degree polynomial c, for which there are polynomials r and s, such that

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

Define the Sylvester matrix of the polynomials p and q

$$\mathscr{R}(p,q) := \begin{bmatrix} p_0 & & & q_0 & & \\ p_1 & p_0 & & q_1 & q_0 & \\ \vdots & p_1 & \ddots & \vdots & q_1 & \ddots \\ p_n & \vdots & \ddots \; p_0 & q_m & \vdots & \ddots \; q_0 \\ & p_n & p_1 & & q_m & q_1 \\ & & \ddots \; \vdots & & & \ddots \; \vdots \\ & & p_n & & & q_m \end{bmatrix} \in \mathbb{R}^{(n+m)\times(n+m)}. \tag{$\mathscr{R}$}$$

(By convention, in this book, all missing entries in a matrix are assumed to be zeros.) A well known fact in algebra is that the degree of the greatest common divisor of p and q is equal to the rank deficiency (co-rank) of $\mathscr{R}(p,q)$, *i.e.*,

$$\text{degree}(c) = n+m-\text{rank}\big(\mathscr{R}(p,q)\big). \tag{GCD}$$

Suppose that p and q have a greatest common divisor of degree $\mathtt{d} > 0$, but the coefficients of the polynomials p and q are imprecise, resulting in perturbed polynomials p_d and q_d. Generically, the matrix $\mathscr{R}(p_\mathrm{d}, q_\mathrm{d})$, constructed from the perturbed polynomials, is full rank, implying that the greatest common divisor of p_d and q_d has degree zero. The problem of finding an *approximate common divisor* of p_d and q_d with degree $\mathtt{d}$, can be formulated as follows. Modify the coefficients of p_d and q_d, as little as possible, so that the resulting polynomials, say, $\widehat{p}$ and $\widehat{q}$ have a greatest common divisor of degree $\mathtt{d}$. This problem is a Sylvester structured low rank approximation problem. Therefore, *Sylvester structured low rank approximation* can be applied for computing an approximate common divisor with a specified degree. The approximate greatest common divisor $\widehat{c}$ for the perturbed polynomials p_d and q_d is the exact greatest common divisor of $\widehat{p}$ and $\widehat{q}$.

The approximate greatest common divisor problem is further discussed in Section 3.2.

Applications in signal processing

Array signal processing

An array of antennas or sensors is used for direction of arrival estimation and adaptive beamforming. Consider $\mathtt{q}$ antennas in a fixed configuration and a wave propagating from distant sources, see Figure 1.3.

First, we consider the case of a single source. The source intensity ℓ_1 (the signal) is a function of time. Let $w(t) \in \mathbb{R}^\mathtt{q}$ be the response of the array at time t (w_i being the response of the ith antenna). Assuming that the source is far from the array (relative to the array's length), the array's response is proportional to the source intensity

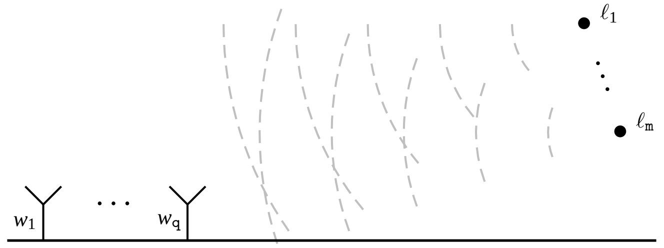

Fig. 1.3 Antenna array processing setup.

$$w(t) = p_1 \ell_1(t-\tau_1),$$

where τ_1 is the time needed for the wave to travel from the source to the array and $p_1 \in \mathbb{R}^\mathtt{q}$ is the array's response to the source emitting at a unit intensity. The vector p_1 depends only on the array geometry and the source location and is therefore constant in time. Measurements of the antenna at time instants $t = 1,\dots,T$ give a data matrix

$$D := \begin{bmatrix} w(1) & \cdots & w(T) \end{bmatrix} = p_1 \underbrace{\begin{bmatrix} \ell_1(1-\tau) & \cdots & \ell_1(T-\tau) \end{bmatrix}}_{\ell_1} = p_1 \ell_1,$$

which has rank equal to one.

Consider now $\mathtt{m} < \mathtt{q}$ distant sources emitting with intensities $\ell_1,\dots,\ell_\mathtt{m}$. Let p_k be the response of the array to the kth source emitting alone with unit intensity. Assuming that the array responds linearly to a mixture of sources, we have

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

where $P := \begin{bmatrix} p_1 & \cdots & p_\mathtt{m} \end{bmatrix}$, $L := \text{col}(\ell_1,\dots,\ell_\mathtt{m})$, and τ_k is the delay of the wave coming from the kth source. This shows that the rank of D is less than or equal to the number of sources $\mathtt{m}$. If the number of sources $\mathtt{m}$ is less than the number of antennas $\mathtt{q}$ and $\mathtt{m}$ is less than the number of samples T, the sources intensities $\ell_1,\dots,\ell_\mathtt{m}$ are linearly independent, and the unit intensity array patterns $p_1,\dots,p_\mathtt{m}$ are linearly independent, then we have that

$$\text{rank}(D) = \text{the number of sources transmitting to the array.}$$

Moreover, the factors P and L in a rank revealing factorization PL of D carry information about the source locations.

With noisy observations, the matrix D is generically a full rank matrix. Then, assuming that the array's geometry is known, low rank approximation can be used to estimate the number of sources and their locations.

A structured low rank approximation algorithm for approximate common divisor computation is presented in Section 3.2.

Applications in chemometrics

Multivariate calibration

A basic problem in chemometrics, called multivariate calibration, is identification of the number and type of chemical components from spectral measurements of mixtures of these components. Let $p_k \in \mathbb{R}^{\mathtt{q}}$ be the spectrum of the kth component at $\mathtt{q}$ predefined frequencies. Under a linearity assumption, the spectrum of a mixture of $\mathtt{m}$ components with concentrations $\ell_1, \dots, \ell_{\mathtt{m}}$ is $d = P\ell$, where $P := \begin{bmatrix} p_1 & \cdots & p_{\mathtt{m}} \end{bmatrix}$ and $\ell = \operatorname{col}(\ell_1, \dots, \ell_{\mathtt{m}})$. Given N mixtures of the components with vectors of concentrations $\ell^{(1)}, \dots, \ell^{(N)}$, the matrix of the corresponding spectra $d_1, \dots, d_N$ is

$$D := \begin{bmatrix} d_1 & \cdots & d_N \end{bmatrix} = \sum_{k=1}^{\mathtt{m}} p_k \underbrace{\begin{bmatrix} \ell_k^{(1)} & \cdots & \ell_k^{(N)} \end{bmatrix}}_{\ell_k} = PL. \tag{RRF}$$

Therefore, the rank of D is less than or equal to the number of components $\mathtt{m}$. Assuming that $\mathtt{q} > \mathtt{m}$, $N > \mathtt{m}$, the spectral responses $p_1, \dots, p_{\mathtt{m}}$ of the components are linearly independent, and the concentrations $\ell_1, \dots, \ell_{\mathtt{m}}$ are linearly independent, we have

$$\operatorname{rank}(D) = \text{the number of chemical components in the mixtures.}$$

The factors P and L in a rank revealing factorization PL of D carry information about the components' spectra and the concentrations of the components in the mixtures. Noisy spectral observations lead to a full rank matrix D, so that low rank approximation can be used to estimate the number of chemical components, their concentrations, and spectra.

Applications in psychometrics

Factor analysis

The psychometric data is test scores and biometrics of a group of people. The test scores can be organized in a data matrix D, whose rows correspond to the scores and the columns correspond to the group members. Factor analysis is a popular method that explains the data as a linear combination of a small number of abilities of the group members. These abilities are called *factors* and the weights by which they are combined in order to reproduce the data are called *loadings*. Factor analysis is based on the assumption that the exact data matrix is low rank with rank being equal to the number of factors. Indeed, the factor model can be written as $D = PL$, where the columns of P correspond to the factors and the rows of L correspond to the loadings. In practice, the data matrix is full rank because the factor model is an idealization of the way test data is generated. Despite the fact that the factor model is

a simplification of the reality, it can be used as an approximation of the way humans perform on tests. Low rank approximation then is a method for deriving optimal in a specified sense approximate psychometric factor models.

The factor model, explained above, is used to assess candidates at the US universities. An important element of the acceptance decision in US universities for undergraduate study is the Scholastic Aptitude Test, and for postgraduate study, the Graduate Record Examination. These tests report three independent scores: writing, mathematics, and critical reading for the Scholastic Aptitude Test; and verbal, quantitative, and analytical for the Graduate Record Examination. The three scores assess what are believed to be the three major factors for, respectively, undergraduate and postgraduate academic performance. In other words, the premise on which the tests are based is that the ability of a prospective student to do undergraduate and postgraduate study is predicted well by a combination of the three factors. Of course, in different areas of study, the weights by which the factors are taken into consideration are different. Even in pure subjects, such as mathematics, however, the verbal as well as quantitative and analytical ability play a role.

> Many graduate-school advisors have noted that an applicant for a mathematics fellowship with a high score on the verbal part of the Graduate Record Examination is a better bet as a Ph.D. candidate than one who did well on the quantitative part but badly on the verbal.
>
> Halmos (1985, page 5)

Applications in machine learning

Natural language processing

Latent semantic analysis is a method in natural language processing for document classification, search by keywords, synonymy and polysemy detection, *etc*. Latent semantic analysis is based on low rank approximation and fits into the pattern of the other methods reviewed here:

1. An exact data matrix is rank deficient with rank related to the complexity of the data generating model.
2. A noisy data matrix is full rank and, for the purpose of approximate modeling, it is approximated by a low rank matrix.

Consider N documents, involving $\mathtt{q}$ terms and $\mathtt{m}$ concepts. If a document belongs to the kth concept only, it contains the ith term with frequency p_{ik}, resulting in the vector of term frequencies $p_k := \operatorname{col}(p_{1k}, \dots, p_{\mathtt{q}k})$, related to the kth concept. The latent semantic analysis model assumes that if a document involves a mixture of the concepts with weights $\ell_1, \dots, \ell_{\mathtt{m}}$ (ℓ_k indicates the relevance of the kth concept to the document), then the vector of term frequencies for that document is

$$d = P\ell, \qquad \text{where} \quad P := \begin{bmatrix} p_1 & \cdots & p_{\mathtt{m}} \end{bmatrix} \quad \text{and} \quad \ell = \operatorname{col}(\ell_1, \dots, \ell_{\mathtt{m}}).$$

Let d_j be the vector of term frequencies, related to the jth document and let $\ell_k^{(j)}$ be the relevance of the kth concepts to the jth document. Then, according to the latent semantic analysis model, the term–document frequencies for the N documents form a data matrix, satisfying (RRF). Therefore, the rank of the data matrix is less than or equal to the number of concepts $\mathtt{m}$. Assuming that $\mathtt{m}$ is smaller than the number of terms $\mathtt{q}$, $\mathtt{m}$ is smaller than the number of documents N, the term frequencies $p_1, \dots, p_{\mathtt{m}}$ are linearly independent, and the relevance of concepts $\ell_1, \dots, \ell_{\mathtt{m}}$ are linearly independent, we have that

$$\operatorname{rank}(D) = \text{the number of concepts related to the documents.}$$

The factors P and L in a rank revealing factorization PL of D carry information about the relevance of the concepts to the documents and the term frequencies related to the concepts.

The latent semantic analysis model is not satisfied exactly in practice because the notion of (small number of) concepts related to (many) documents is an idealization. Also the linearity assumption is not likely to hold in practice. In reality the term–document frequencies matrix D is full rank indicating that the number of concepts is equal to either the number of terms or the number of documents. Low rank approximation, however, can be used to find a small number of concepts that explain approximately the term–documents frequencies via the model (RRF). Subsequently, similarity of documents can be evaluated in the concepts space, which is a low dimensional vector space. For example, the j_1th and j_2th documents are related if they have close relevance $\ell_k^{(j_1)}$ and $\ell_k^{(j_2)}$ to all concepts $k = 1, \dots, \mathtt{m}$. This gives a way to classify of the documents. Similarly, terms can be clustered in the concepts space by looking at the rows of the P matrix. Nearby rows of P correspond to terms that are related to the same concepts. (Such terms are likely to be synonymous.) Finally, a search for documents by keywords can be done by first translating the keywords to a vector in the concepts space and then finding a nearby cluster of documents to this vector. For example, if there is a single keyword, which is the ith term, then the ith row of the P matrix shows the relevant combination of concepts for this search.

Recommender system

The main issue underlying the abstract low rank approximation problem and the applications reviewed up to now is data approximation. In the recommender system problem, the main issue is the one of *missing data*: given ratings of some items by some users, infer the missing ratings. Unique recovery of the missing data is impossible without additional assumptions. The underlying assumption in many recommender system problems is that the complete matrix of the ratings is of low rank.

Consider $\mathtt{q}$ items and N users and let d_{ij} be the rating of the ith item by the jth user. As in the psychometrics example, it is assumed that there is a "small" number $\mathtt{m}$ of "typical" (or characteristic, or factor) users, such that all user ratings can be

obtained as linear combinations of the ratings of the typical users. This implies that the complete matrix $D = \begin{bmatrix} d_{ij} \end{bmatrix}$ of the ratings has rank $\mathtt{m}$, *i.e.*,

$$\operatorname{rank}(D) = \text{number of "typical" users.}$$

Then exploiting the prior knowledge that the number of "typical" users is small, the missing data recovery problem can be posed as the following matrix completion problem

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

This gives a procedure for solving the exact modelling problem (the given elements of D are assumed to be exact). The corresponding solution method can be viewed as the equivalent of the rank revealing factorization problem in exact modeling problems, for the case of complete data.

Of course, the rank minimization problem (MC) is much harder to solve than the rank revealing factorization problem. Moreover, theoretical justification and additional assumptions (about the number and distribution of the given elements of D) are needed for a solution $\widehat{D}$ of (MC) to be unique and to coincide with the complete true matrix D. It turns out, however, that under certain specified assumptions exact recovery is possible by solving the convex optimization problem obtained by replacing $\operatorname{rank}(\widehat{D})$ in (MC) with the nuclear norm

$$\|\widehat{D}\|_* := \text{sum of the singular values of } \widehat{D}.$$

The importance of the result is that under the specified assumptions the hard problem (MC) can be solved efficiently and reliably by convex optimization methods.

In real-life application of recommender systems, however, the additional problem of data approximation occurs. In this case the constraint $\widehat{D}_{ij} = D_{ij}$ of (MC) has to be relaxed, *e.g.*, replacing it by

$$\widehat{D}_{ij} = D_{ij} + \Delta D_{ij},$$

where ΔD_{ij} are corrections, accounting for the data uncertainty. The corrections are additional optimization variables. Taking into account the prior knowledge that the corrections are small, a term $\lambda \|\Delta D\|_{\mathrm{F}}$ is added in the cost function. The resulting matrix approximation problem is

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

In a stochastic setting the term $\lambda \|\Delta D\|_{\mathrm{F}}$ corresponds to the assumption that the true data D is perturbed with noise that is zero mean, Gaussian, independent, and with equal variance.

Again the problem can be relaxed to a convex optimization problem by replacing rank with nuclear norm. The choice of the γ parameter reflects the trade-off between complexity (number of identified "typical" users) and accuracy (size of the correction ΔD) and depends in the stochastic setting on the noise variance.

Nuclear norm and low rank approximation methods for estimation of missing values are developed in Sections 3.3 and 5.1.

Multidimensional scaling

Consider a set of N points in the plane

$$\mathscr{X} := \{x_1, \dots, x_N\} \subset \mathbb{R}^2$$

and let d_{ij} be the squared distance from x_i to x_j, *i.e.*,

$$d_{ij} := \|x_i - x_j\|_2^2.$$

The $N \times N$ matrix $D = \begin{bmatrix} d_{ij} \end{bmatrix}$ of the pair-wise distances, called in what follows the distance matrix (for the set of points $\mathscr{X}$), has rank at most 4. Indeed,

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

so that

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

The localization problem from pair-wise distances is: given the distance matrix D, find the locations $\{x_1, \dots, x_N\}$ of the points up to a rigid transformation, *i.e.*, up to translation, rotation, and reflection of the points. Note that rigid transformations preserve the pair-wise distances, so that the distance matrix D alone is not sufficient to locate the points uniquely.

With exact data, the problem can be posed and solved as a rank revealing factorization problem $(*)$. With noisy measurements, however, the matrix D is generically full rank. In this case, the relative (up to rigid transformation) point locations can be *estimated* by approximating D by a rank-4 matrix $\widehat{D}$. In order to be a valid distance matrix, however, $\widehat{D}$ must have the structure

$$\widehat{D} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \begin{bmatrix} \widehat{x}_1^\top \widehat{x}_1 & \cdots & \widehat{x}_N^\top \widehat{x}_N \end{bmatrix} - 2\widehat{X}^\top \widehat{X} + \begin{bmatrix} \widehat{x}_1^\top \widehat{x}_1 \\ \vdots \\ \widehat{x}_N^\top \widehat{x}_N \end{bmatrix} \begin{bmatrix} 1 & \cdots & 1 \end{bmatrix}, \qquad \text{(MDS)}$$

for some $\widehat{X} = \begin{bmatrix} \widehat{x}_1 & \cdots & \widehat{x}_N \end{bmatrix}$, *i.e.*, the estimation problem is a *bilinearly structured low rank approximation problem.*

Microarray data analysis

The measurements of a microarray experiment are collected in a $\mathtt{q} \times N$ real matrix D—rows correspond to genes and columns correspond to time instances. The element d_{ij} is the *expression level* of the ith gene at the jth moment of time. The rank of D is equal to the number of *transcription factors* that regulate the gene expression levels

$$\operatorname{rank}(D) = \text{number of transcription factors.}$$

In a rank revealing factorization $D = PL$, the jth column of L is a vector of intensities of the transcription factors at time j, and the ith row of P is a vector of sensitivities of the ith gene to the transcription factors. For example, p_{ij} equal to zero means that the jth transcription factor does not regulate the ith gene.

An important problem in bioinformatics is to discover what transcription factors regulate a particular gene and what their time evaluations are. This problem amounts to computing an (approximate) factorization PL of the matrix D. The need of approximation comes from:

1. inability to account for all relevant transcription factors (therefore accounting only for a few dominant ones), and
2. measurement errors occurring in the collection of the data.

Often it is known a priori that certain transcription factors do not regulate certain genes. This implies that certain elements of the sensitivity matrix P are known to be zeros. In addition, the transcription factor activities are modeled to be nonnegative, smooth, and periodic functions of time. Where transcription factors down regulate a gene, the elements of P have to be negative to account for this. In Section 5.4, this prior knowledge is formalized as constraints in a low rank matrix approximation problem and optimization methods for solving the problem are developed.

Applications in computer vision

Conic section fitting

In the applications reviewed so far, the low rank approximation problem was applied to *linear* data modeling. Nonlinear data modeling, however, can also be formulated as a low rank approximation problem. The key step "linearizing" the problem is preprocessing of the data by the nonlinear function, defining the model structure. In the machine learning literature, where nonlinear data modeling is a common practice, the nonlinear function is called the *feature map* and the resulting modeling methods are referred to as *kernel methods*.

As a specific example, consider the problem of fitting data by a conic section, *i.e.*, given a set of points in the plane

$$\{d_1,\dots,d_N\}\subset\mathbb{R}^2,\qquad \text{where}\quad d_j=\operatorname{col}(x_j,y_j),$$

find a conic section

$$\mathscr{B}(A,b,c):=\{\,d\in\mathbb{R}^2 \mid d^\top A d + b^\top d + c = 0\,\}. \tag{$\mathscr{B}(A,b,c)$}$$

that fits them. Here A is a 2×2 symmetric matrix, b is a 2×1 vector, and c is a scalar. A, b, and c are parameters defining the conic section. In order to avoid a trivial case $\mathscr{B}=\mathbb{R}^2$, we assume that at least one of the parameters A, b, or c is nonzero. The representation ($\mathscr{B}(A,b,c)$) is called implicit representation, because it imposes a relation (implicit function) on the elements x and y of d.

Defining the parameter vector

$$\theta := \begin{bmatrix} a_{11} & 2a_{12} & b_1 & a_{22} & b_2 & c \end{bmatrix},$$

and the extended data vector

$$d_{\text{ext}} := \operatorname{col}(x^2, xy, x, y^2, y, 1), \tag{d_{ext}}$$

we have

$$d\in\mathscr{B}(\theta)=\mathscr{B}(A,b,c) \qquad\iff\qquad \theta d_{\text{ext}} = 0.$$

(In the machine learning terminology, the map $d\mapsto d_{\text{ext}}$, defined by (d_{ext}), is the feature map for the conic section model.) Consequently, all data points $d_1,\dots,d_N$ are fitted by the model if

$$\theta \underbrace{\begin{bmatrix} d_{\text{ext},1} & \cdots & d_{\text{ext},N}\end{bmatrix}}_{D_{\text{ext}}} = 0 \qquad\iff\qquad \operatorname{rank}(D_{\text{ext}})\le 5. \tag{CSF}$$

Therefore, for $N>5$ data points, exact fitting is equivalent to rank deficiency of the extended data matrix D_{ext}. For $N<5$ data points, there is nonunique exact fit independently of the data. For $N=5$ different points, the exact fitting conic section is unique, see Figure 1.4.

With $N>5$ noisy data points, the extended data matrix D_{ext} is generically full rank, so that an exact fit does not exists. A problem called geometric fitting is to minimize the sum of squared distances from the data points to the conic section. The problem is equivalent to *quadratically structured low rank approximation*.

Generalization of the conic section fitting problem to algebraic curve fitting and solution methods are presented in Chapter 6.

Exercise 1.1. Find and plot another conic section that fits the points

$$d_1=\begin{bmatrix}0.2\\0.2\end{bmatrix},\quad d_2=\begin{bmatrix}0.8\\0.2\end{bmatrix},\quad d_3=\begin{bmatrix}0.2\\0.8\end{bmatrix},\quad \text{and}\quad d_4=\begin{bmatrix}0.8\\0.8\end{bmatrix},$$

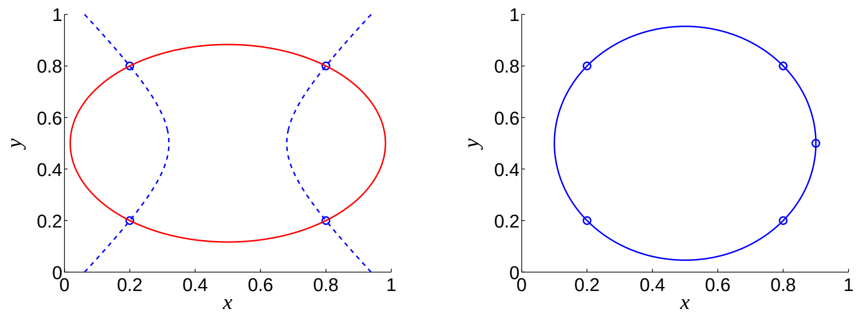

Fig. 1.4 Conic section fitting. Left: $N=4$ points (circles) have nonunique fit (two fits are shown in the figure with solid and dashed lines). Right: $N=5$ different points have a unique fit (solid line).

in Figure 1.4, left. □

Exercise 1.2. Find the parameters $(A,b,1)$ in the representation ($\mathscr{B}(A,b,c)$) of the ellipse in Figure 1.4, right. The data points are:

$$d_1=\begin{bmatrix}0.2\\0.2\end{bmatrix},\quad d_2=\begin{bmatrix}0.8\\0.2\end{bmatrix},\quad d_3=\begin{bmatrix}0.2\\0.8\end{bmatrix},\quad d_4=\begin{bmatrix}0.8\\0.8\end{bmatrix},\quad \text{and}\quad d_5=\begin{bmatrix}0.9\\0.5\end{bmatrix}, \qquad \square$$

(Answer: $A=\left[\begin{smallmatrix}3.5156 & 0\\ 0 & 2.7344\end{smallmatrix}\right]$, $b=\left[\begin{smallmatrix}-3.5156\\ -2.7344\end{smallmatrix}\right]$)

Fundamental matrix estimation

A scene is captured by two cameras at fixed locations (stereo vision) and N matching pairs of points

$$\{u_1,\dots,u_N\}\subset\mathbb{R}^2 \qquad\text{and}\qquad \{v_1,\dots,v_N\}\subset\mathbb{R}^2$$

are located in the resulting images. Corresponding points u and v of the two images satisfy what is called an *epipolar constraint*

$$\begin{bmatrix} v^\top & 1\end{bmatrix} F \begin{bmatrix} u \\ 1\end{bmatrix} = 0, \qquad \text{for some } F\in\mathbb{R}^{3\times 3}, \text{ with } \operatorname{rank}(F)=2. \tag{EPI}$$

The 3×3 matrix $F\neq 0$, called the *fundamental matrix*, characterizes the relative position and orientation of the cameras and does not depend on the selected pairs of points. Estimation of F from data is a necessary calibration step in many computer vision methods.

The epipolar constraint (EPI) is linear in F. Indeed, defining

$$d_{\text{ext}} := \begin{bmatrix} u_xv_x & u_xv_y & u_x & u_yv_x & u_yv_y & u_y & v_x & v_y & 1\end{bmatrix}^\top \in\mathbb{R}^9, \tag{d'_{ext}}$$

where $u = \operatorname{col}(u_x, u_y)$ and $v = \operatorname{col}(v_x, v_y)$, (EPI) can be written as

$$\operatorname{vec}^\top(F) d_{\text{ext}} = 0.$$

Note that, as in the application for conic section fitting, the original data (u, v) is mapped to an extended data vector d_{ext} via a nonlinear function (a feature map). In this case, however, the function is *bilinear*.

Taking into account the epipolar constraints for all data points, we obtain the matrix equation

$$\operatorname{vec}^\top(F) D_{\text{ext}} = 0, \qquad \text{where} \quad D_{\text{ext}} := \begin{bmatrix} d_{\text{ext},1} & \cdots & d_{\text{ext},N} \end{bmatrix}. \tag{FME}$$

The rank constraint imposed on F implies that F is a nonzero matrix. Therefore, by (FME), for $N \geq 8$ data points, the extended data matrix D_{ext} is rank deficient. Moreover, the fundamental matrix F can be reconstructed up to a scaling factor from a vector in the left kernel of D_{ext}.

Noisy data with $N \geq 8$ data points generically gives rise to a full rank extended data matrix D_{ext}. The estimation problem is a *bilinearly structured low rank approximation* problem with an additional constraint that $\operatorname{rank}(F) = 2$.

Summary of applications

Table 1.1 summarized the reviewed applications: given data, data matrix constructed from the original data, structure of the data matrix, and meaning of the rank of the data matrix in the context of the application. More applications are mentioned in the notes and references section in the end of the chapter.

1.4 Overview of algorithms

The rank constraint in the low rank approximation problem corresponds to the constraint in the data modeling problem that the data is fitted exactly by a linear model of bounded complexity. Therefore, the question of representing the rank constraint in the low rank approximation problem corresponds to the question of choosing the model representation in the data fitting problem. Different representations lead to

- *optimization problems*, the relation among which may not be obvious;
- *algorithms*, which may have different convergence properties and efficiency; and
- *numerical software*, which may have different numerical robustness.

The virtues of the abstract, representation free, low rank approximation problem formulation, are both *conceptual:* it clarifies the equivalence among different parameter optimization problems, and *practical:* it shows various ways of formulating one and the same high level data modeling problem as parameter optimization problems. On

Table 1.1 Meaning of the rank of the data matrix in the applications.

application	data	data matrix	structure	rank =	ref.
approximate realization	impulse response H	$\mathscr{H}(H)$	Hankel	system's order	Sec. 2.2
stochastic realization	autocorrelation function R	$\mathscr{H}(R)$	Hankel	system's order	—
system identification	trajectory w of the system	$\mathscr{H}_{\mathrm{n}_{\max}+1}(w)$	Hankel	(SYSID)	Sec. 4.3
approximate GCD	polynomials p_{d} and q_{d}	$\mathscr{R}(p_{\mathrm{d}}, q_{\mathrm{d}})$	Sylvester	(GCD)	Sec. 3.2
array processing	array response $\big(w(1), \dots, w(T)\big)$	$\begin{bmatrix} w(1) & \cdots & w(T) \end{bmatrix}$	unstructured	# of signal sources	—
multivariate calibration	spectral responses $\{d_1, \dots, d_N\} \subset \mathbb{R}^{\mathrm{q}}$	$\begin{bmatrix} d_1 & \cdots & d_N \end{bmatrix}$	unstructured	# of chemical components	—
factor analysis	test scores d_{ij}	$[d_{ij}]$	unstructured	# of factors	—
natural language processing	term–document frequencies d_{ij}	$[d_{ij}]$	unstructured	# of concepts	—
recommender system	some ratings d_{ij}	$[d_{ij}]$	unstructured missing data	# of tastes	Sec. 5.1
multidimansional scaling	pair-wise distances d_{ij}	$[d_{ij}]$	(MDS)	$\dim(x) + 2$	—
microarray data analysis	gene expression levels d_{ij}	$[d_{ij}]$	unstructured	# of transcript. factors	Sec. 5.4
conic section fitting	points $\{d_1, \dots, d_N\} \subset \mathbb{R}^2$	(d_{ext}), (CSF)	quadratic	5	Ch. 6
fundamental matrix estimation	points $u_j, v_j \in \mathbb{R}^2$	(d'_{ext}), (FME)	bilinear	6	—

the conceptual level, the low rank approximation problem formulation shows what one aims to achieve without a reference to implementation details. In particular, the model representation is such a detail, which is not needed for a high level formulation of data modeling problems. As discussed next, however, the representation is unavoidable when one solves the problem analytically or numerically.

On the practical level, the low rank problem formulation allows one to translate the abstract data modeling problem to different concrete parametrized problems by choosing the model representation. Different representations naturally lend themselves to different analytical and numerical methods. For example, a controllable linear time-invariant system can be represented by a transfer function, state space, convolution, *etc*. representations. The analysis tools related to these representations are rather different and, consequently, the obtained solutions differ despite of the fact that they solve the same abstract problem. Moreover, the parameter optimization problems, resulting from different model representations, lead to algorithms and numerical implementations, whose robustness properties and computational efficiency differ. Although, often in practice, there is no universally "best" algorithm or software implementation, having a wider set of available options is an advantage.

Independent of the choice of the rank representation only a few special low rank approximation problems have analytic solutions. So far, the most important special case with an analytic solution is the unstructured low rank approximation in the Frobenius norm. The solution in this case can be obtained from the singular value decomposition of the data matrix (Eckart–Young–Mirsky theorem). Extensions of this basic solution are problems known as generalized and restricted low rank approximation, where some columns or, more generally submatrices of the approximation, are constrained to be equal to given matrices. The solutions to these problems are given by, respectively, the generalized and restricted singular value decompositions. Another example of low rank approximation problem with analytical solution is the circulant structured low rank approximation, where the solution is expressed in terms of the discrete Fourier transform of the data.

In general, low rank approximation problems are NP-hard. There are three fundamentally different solution approaches for the general low rank approximation problem:

- heuristic methods based on convex relaxations,
- local optimization methods, and
- global optimization methods.

From the class of heuristic methods the most popular ones are the subspace methods. The approach used in the subspace type methods is to relax the difficult low rank approximation problem to a problem with an analytic solution in terms of the singular value decomposition, *e.g.*, ignore the structure constraint of a structured low rank approximation problem. The subspace methods are found to be very effective in model reduction, system identification, and signal processing. The class of the subspace system identification methods is based on the unstructured low rank approximation in the Frobenius norm (*i.e.*, singular value decomposition) while the original problems are Hankel structured low rank approximation.

The methods based on local optimization split into two main categories:

- *alternating projections* and
- *variable projections*

type algorithms. Both alternating projections and variable projections exploit the bilinear structure of the low rank approximation problems.

In order to explain the ideas underlining the alternating projections and variable projections methods, consider the optimization problem

$$\text{minimize} \quad \text{over } P \in \mathbb{R}^{\mathtt{q}\times\mathtt{m}} \text{ and } L \in \mathbb{R}^{\mathtt{m}\times N} \quad \|D - PL\|_{\mathrm{F}}^2 \tag{LRA$_P$}$$

corresponding to low rank approximation with an image representation of the rank constraint. The term PL is bilinear in the optimization variables P and L, so that for a fixed P, (LRA$_P$) becomes a linear least squares problem in L and vice verse,

for a fixed L, (LRA$_P$) becomes a linear least squares problem in P. This suggests an iterative algorithm starting from an initial guess for P and L and alternatively solves the problem with one of the variables fixed. Since each step is a projection operation the method has the name alternating projections. It is globally convergent to a locally optimal solution of (LRA$_P$) with a linear convergence rate.

The bilinear nature of (LRA$_P$) implies that for a fixed P the problem can be solved in closed form with respect to L. This gives us an equivalent cost function depending on P. Subsequently, the original problem can be solved by minimizing the equivalent cost function over P. Of course, the latter problem is a nonlinear optimization problem. Standard local optimization methods can be used for this purpose. The elimination of the L variable from the problem has the advantage of reducing the number of optimization variables, thus simplifying the problem. Evaluation of the cost function for a given P is a projection operation. In the course of the nonlinear minimization over P, this variable changes, thus the name of the method—variable projections.

In the statistical literature, the alternating projections algorithm is given the interpretation of *expectation maximization*. The problem of computing the optimal approximation $\widehat{D} = PL$, given P is the expectation step and the problem of computing P, given L is the maximization step of the expectation maximization procedure.

1.5 Literate programming

At first, I thought programming was primarily analogous to musical composition—to the creation of intricate patterns, which are meant to be performed. But lately I have come to realize that a far better analogy is available: Programming is best regarded as the process of creating *works of literature*, which are meant to be read.

Knuth (1992, page ix)

The ideas presented in the book are best expressed as algorithms for solving data modeling problems. The algorithms, in turn, are practically useful when implemented in ready-to-use software. The gap between the theoretical discussion of data modeling methods and the practical implementation of these methods is bridged by using a literate programming style. The software implementation (MATLAB code) is interwoven in the text, so that the full implementation details are available in a human readable format and they come in the appropriate context of the presentation.

A literate program is composed of interleaved code segments, called chunks, and text. The program can be split into chunks in any way and the chunks can be presented in any order, deemed helpful for the understanding of the program. This allows us to focus on the logical structure of the program rather than the way a computer executes it. The actual computer executable code is *tangled* from a *web* of the code chunks by skipping the text and putting the chunks in the right order. In addition, literate programming allows us to use a powerful typesetting system such as LaTeX (rather than plain text) for the documentation of the code.

We use the `noweb` system for literate programming. Its main advantage over alternative systems is independence of the programming language being used. Next, the typographic conventions needed to follow the presentation are explained.

The code is typeset in `small true type` font and consists of a number of code chunks. The code chunks begin with tags enclosed in angle brackets (*e.g.*, ⟨*code tag*⟩) and are sequentially numbered by the page number and a letter identifying them on the page. Thus the chunk's identification number (given to the left of the tag) is also used to locate the chunk in the text.

For example,

25a ⟨*Print a figure* 25a⟩≡
```
function print_fig(file_name)
xlabel('x'), ylabel('y'), title('t'), set(gca, 'fontsize', 25)
eval(sprintf('print -depsc %s.eps', file_name));
```
Defines:
`print_fig`, used in chunks 100c, 111b, 114b, 125e, 127d, 163b, 194b, 215c, 216a, and 223c.

(a function exporting the current figure to an encapsulated postscript file with a specified name, using default labels and font size) has identification number **??**, locating the code as being on page **??**.

If a chunk is included in, is a continuation of, or is continued by other chunk(s), its definition has references to the related chunk(s). The sintax convention for doing this is best explained by an example.

Example: block Hankel matrix constructor

Consider the implementation of the (block) Hankel matrix constructor

$$\mathscr{H}_{i,j}(w) := \begin{bmatrix} w(1) & w(2) & \cdots & w(j) \\ w(2) & w(3) & \cdots & w(j+1) \\ \vdots & \vdots & & \vdots \\ w(i) & w(i+1) & \cdots & w(j+i-1) \end{bmatrix}, \qquad (\mathscr{H}_{i,j})$$

where

$$w = \big(w(1), \dots, w(T)\big), \qquad \text{with} \quad w(t) \in \mathbb{R}^{\mathtt{q} \times N}.$$

The definition of the function, showing its input and output arguments, is

25b ⟨*Hankel matrix constructor* 25b⟩≡
```
function H = blkhank(w, i, j)
```
Defines:
`blkhank`, used in chunks 25, 77b, 80a, 88a, 109a, 113, 115a, 117a, 119, 124b, 126b, and 213.

(The reference to the right of the identification tag shows that the definition is continued in chunk number **??**.) The third input argument of `blkhank`—the number of block columns `j` is optional. Its default value is maximal number of block columns

$$j = T - i + 1.$$

26a ⟨*optional number of (block) columns* 26a⟩≡
```
if nargin < 3 | isempty(j), j = T - i + 1; end
if j <= 0, error('Not enough data.'), end
```

(The reference to the right of the identification tag now shows that this chunk is included in other chunks.)

Two cases are distinguished, depending on whether w is a vector ($N = 1$) or matrix ($N > 1$) valued trajectory.

26b ⟨*Hankel matrix constructor* 25b⟩+≡
```
if length(size(w)) == 3
  ⟨matrix valued trajectory w 26c⟩
else
  ⟨vector valued trajectory w 26f⟩
end
```

(The reference to the right of the identification tag shows that this chunk is a continuation of chunk **??** and is not followed by other chunks. Its body includes chunks **??** and **??**.)

- If w is a matrix valued trajectory, the input argument `w` should be a 3 dimensional tensor, constructed as follows:

$$\mathtt{w}(:,:,\mathtt{t}) = w(t)$$

26c ⟨*matrix valued trajectory w* 26c⟩≡
```
[ttw, N, T]  = size(w);
⟨optional number of (block) columns 26a⟩
```

(This chunk is both included and followed by other chunks.) In this case, the construction of the block Hankel matrix $H_{i,j}(w)$ is done explicitly by a double loop:

26d ⟨*matrix valued trajectory w* 26c⟩+≡
```
H = zeros(i * ttw, j * N);
for ii = 1:i
  for jj = 1:j
    H(((ii - 1) * ttw + 1):(ii * ttw), ...
      ((jj - 1) * N + 1):(jj * N)) = w(: ,:, ii + jj - 1);
  end
end
```

- If w is a vector valued trajectory , the input argument `w` should be a matrix formed as follows $\mathtt{w}(:,\mathtt{t}) = w(t)$, however, since T must be greater than or equal to the number of variables $\mathtt{q} := \dim\big(w(t)\big)$, when `w` has more rows than columns, the input is treated as $\mathtt{w}(\mathtt{t},:) = w^\top(t)$.

26e ⟨*reshape w and define* q, *T* 26e⟩≡
```
[ttw, T] = size(w);
if T < ttw, w = w'; [ttw, T] = size(w); end
```

26f ⟨*vector valued trajectory w* 26f⟩≡
```
⟨reshape w and define q, T 26e⟩
⟨optional number of (block) columns 26a⟩
```

The reason to consider the case of a vector valued w separately is that in this case the construction of the Hankel matrix $\mathscr{H}_{i,j}(w)$ can be done with a single loop along the block rows.

27 ⟨*vector valued trajectory w* 26f⟩+≡

```
H = zeros(i * ttw, j);
for ii = 1:i
  H(((ii - 1) * ttw + 1):(ii * ttw), :) = ...
                          w(:, ii:(ii + j - 1));
end
```

Since in typical situations when `blkhank` is used (system identification problems), $i \ll j$ and MATLAB being an interpreted language executes `for` loops slowly, the reduction to a single `for` loop along the block rows of the matrix leads to significant decrease of the execution time compared to the implementation with two `for` loops in the general case.

Exercise 1.3. Download and install `noweb` from

`www.cs.tufts.edu/~nr/noweb/` □

Exercise 1.4. Write a literate program for constructing the Sylvester matrix $\mathscr{R}(p,q)$, defined in ($\mathscr{R}$) on page 11. Use your program to find the degree `d` of the greatest common divisor of the polynomials

$$p(z) = 1 + 3z + 5z^2 + 3z^3 \qquad \text{and} \qquad q(z) = 3 + 5z + 3z^2 + z^3. \qquad \square$$

(Answer: $\texttt{d} = 1$)

1.6 Notes and references

Classical and behavioural paradigms for data modeling

Methods for solving overdetermined systems of linear equations (*i.e.*, data modeling methods using the classical input/output representation paradigm) are reviewed in Appendix A. The behavioral paradigm for data modeling was put forward by Jan C. Willems in the early 80's. It became firmly established with the publication of the three part paper (Willems, 1986, 1987). Other landmark publications on the behavioral paradigm are (Willems, 1989, 1991, 2007), and the book (Polderman and Willems, 1998).

The numerical linear algebra problem of low rank approximation is a computational tool for data modeling, which fits the behavioral paradigm as "a hand fits a glove". Historically the low rank approximation problem is closely related to the singular value decomposition, which is a method for computing low rank approximations and is a main tool in many algorithms for data modeling. A historical account of the development of the singular value decomposition is given in (Stewart, 1993). The Eckart–Young–Mirsky matrix low rank approximation theorem is proven in (Eckart and Young, 1936).

Applications

For details about the realization and system identification problems, see Sections 2.2, 3.1, and 4.3. Direction of arrival and adaptive beamforming problems are discussed in (Krim and Viberg, 1996; Kumaresan and Tufts, 1983). Low rank approximation methods (alternating least squares) for estimation of mixture concentrations in chemometrics are proposed in (Wentzell et al, 1997). An early reference on the approximate greatest common divisor problem is (Karmarkar and Lakshman, 1998). Efficient optimization based methods for approximate greatest common divisor computation are discussed in Section 3.2. Other computer algebra problems that reduce to structured low rank approximation are discussed in (Botting, 2004).

Many problems for information retrieval in machine learning, see, *e.g.*, (Bishop, 2006; Fierro and Jiang, 2005; Shawe-Taylor and Cristianini, 2004), are low rank approximation problems and the corresponding solution techniques developed in the machine learning community are methods for solving low rank approximation problems. For example, clustering problems have been related to low rank approximation problems in (Ding and He, 2004; Kiers, 2002; Vichia and Saporta, 2009). Machine learning problems, however, are often posed in a stochastic estimation setting which obscures their deterministic approximation interpretation. For example, principal component analysis (Jackson, 2003; Jolliffe, 2002) and unstructured low rank approximation with Frobenius norm are equivalent optimization problems. The principal component analysis problem, however, is motivated in a statistical setting and for this reason may be considered as a different problem. In fact, principal component analysis provides another (statistical) interpretation of the low rank approximation problem.

The conic section fitting problem has extensive literature, see Chapter 6 and the tutorial paper (Zhang, 1997). The kernel principal component analysis method is developed in the machine learning and statistics literature (Schölkopf et al, 1999). Despite of the close relation between kernel principal component analysis and conic section fitting, the corresponding literature are disjoint. Closely related to the estimation of the fundamental matrix problem in two-view computer vision is the shape from motion problem (Ma et al, 2004; Tomasi and Kanade, 1993).

Matrix factorization techniques have been used in the analysis of microarray data in (Alter and Golub, 2006) and (Kim and Park, 2007). Alter and Golub (2006) propose a principal component projection to visualize high dimensional gene expression data and show that some known biological aspects of the data are visible in a two dimensional subspace defined by the first two principal components.

Distance problems

The low rank approximation problem aims at finding the "smallest" correction of a given matrix that makes the corrected matrix rank deficient. This is a special case of a distance problems: find the "nearest" matrix with a specified property to a given matrix. For an overview of distance problems, see (Higham, 1989). In (Byers, 1988), an algorithm for computing the distance of a stable matrix (Hurwitz matrix in the continuous-time case and Schur matrix in the discrete-time case) to the set of unstable matrices is presented. Stability radius for structured perturbations and its relation to the algebraic Riccati equation is presented in (Hinrichsen and Pritchard, 1986).

Structured linear algebra

Related to the topic of distance problems is the grand idea that the whole linear algebra (solution of systems of equations, matrix factorization, *etc*.) can be generalized to uncertain data. The uncertainty is described as structured perturbation on the data and a solution of the problem is obtained by correcting the data with a correction of the smallest size that renders the problem solvable for the corrected data. One of the first references on the topic of structured linear algebra are (Chandrasekaran et al, 1998; El Ghaoui and Lebret, 1997).

Structured pseudospectra

Let $\lambda(A)$ be the set of eigenvalues of $A \in \mathbb{C}^{n\times n}$ and $\mathscr{M}$ be a set of structured matrices

$$\mathscr{M} := \{\, \mathscr{S}(p) \mid p \in \mathbb{R}^{n_p} \,\},$$

with a given structure specification $\mathscr{S}$. The ε-structured pseudospectra (Graillat, 2006; Trefethen and Embree, 1999) of A is defined as the set

$$\lambda_\varepsilon(A) := \{\, z \in \mathbb{C} \mid z \in \lambda(\widehat{A}),\ \widehat{A} \in \mathscr{M},\ \text{and } \|A-\widehat{A}\|_2 \le \varepsilon \,\}.$$

Using the structured pseudospectra, one can determine the structured distance of A to singularity as the minimum of the following optimization problem:

$$\text{minimize} \quad \text{over } \widehat{A} \quad \|A-\widehat{A}\|_2 \quad \text{subject to} \quad \widehat{A} \text{ is singular and } \widehat{A} \in \mathscr{M}.$$

This is a special structured low rank approximation problem for squared data matrix and rank reduction by one. Related to structured pseudospectra is the structured condition number problem for a linear system of equations, see (Rump, 2003).

Statistical properties

Related to low rank approximation are the *orthogonal regression* (Gander et al, 1994), *errors-in-variables* (Gleser, 1981), and *measurement errors* methods in the statistical literature (Carroll et al, 1995; Cheng and Van Ness, 1999). Classic papers on the univariate errors-in-variables problem are (Adcock, 1877, 1878; Koopmans, 1937; Madansky, 1959; Pearson, 1901; York, 1966). Closely related to the errors-

in-variables framework for low rank approximation is the probabilistic principal component analysis framework of (Tipping and Bishop, 1999).

Reproducible research

“An article about computational science in a scientific publication is not the scholarship itself, it is merely advertising of the scholarship. The actual scholarship is the complete software development environment and the complete set of instructions which generated the figures.” Buckheit and Donoho (1995)

The reproducible research concept is at the core of all sciences. In applied fields such as data modeling, however, algorithms’ implementation, availability of data, and reproducibility of the results obtained by the algorithms on data are often neglected. This leads to a situation, described in (Buckheit and Donoho, 1995) as a scandal. See also (Kovacevic, 2007).

A quick and easy way of making computational results obtained in MATLAB reproducible is to use the function `publish`. Better still, the code and the obtained results can be presented in a literate programming style.

Literate programming

The creation of the literate programming is a byproduct of the TEX project, see (Knuth, 1984, 1992). The original system, called `web` is used for documentation of the TEX program (Knuth, 1986) and is for the Pascal language. Later a version `cweb` for the C language was developed. The `web` and `cweb` systems are followed by many other systems for literate programming that target specific languages. Unfortunately this leads to numerous literate programming dialects.

The `noweb` system for literate programming, created by N. Ramsey in the mid 90’s, is not bound to any specific programming language and text processing system. A tutorial introduction is given in (Ramsey, 1994). The `noweb` syntax is also adopted in the babel part of Emacs `org-mode` (Dominik, 2010)—a package for keeping structured notes that includes support for organization and automatic evaluation of computer code.

References

Adcock R (1877) Note on the method of least squares. The Analyst 4:183–184
Adcock R (1878) A problem in least squares. The Analyst 5:53–54
Alter O, Golub GH (2006) Singular value decomposition of genome-scale mRNA lengths distribution reveals asymmetry in RNA gel electrophoresis band broadening. Proceedings of the National Academy of Sciences 103:11,828–11,833
Bishop C (2006) Pattern recognition and machine learning. Springer

Botting B (2004) Structured total least squares for approximate polynomial operations. Master's thesis, School of Computer Science, University of Waterloo

Buckheit J, Donoho D (1995) Wavelets and statistics, Springer-Verlag, Berlin, New York, chap "Wavelab and reproducible research"

Byers R (1988) A bisection method for measuring the distance of a stable matrix to the unstable matrices. SIAM J Sci Stat Comput 9(5):875–881

Carroll R, Ruppert D, Stefanski L (1995) Measurement Error in Nonlinear Models. Chapman & Hall/CRC, London

Chandrasekaran S, Golub G, Gu M, Sayed A (1998) Parameter estimation in the presence of bounded data uncertainties. SIAM J Matrix Anal Appl 19:235–252

Cheng C, Van Ness JW (1999) Statistical regression with measurement error. London: Arnold

Ding C, He X (2004) K-means clustering via principal component analysis. In: Proc. of Int. Conf. Machine Learning, pp 225–232

Dominik C (2010) The org mode 7 reference manual. Network theory ltd, URL `http://orgmode.org/`

Eckart G, Young G (1936) The approximation of one matrix by another of lower rank. Psychometrika 1:211–218

El Ghaoui L, Lebret H (1997) Robust solutions to least-squares problems with uncertain data. SIAM J Matrix Anal Appl 18:1035–1064

Fierro R, Jiang E (2005) Lanczos and the Riemannian SVD in information retrieval applications. Numer Linear Algebra Appl 12:355–372

Gander W, Golub G, Strebel R (1994) Fitting of circles and ellipses: Least squares solution. BIT 34:558–578

Gleser L (1981) Estimation in a multivariate "errors in variables" regression model: large sample results. The Annals of Statistics 9(1):24–44

Graillat S (2006) A note on structured pseudospectra. J of Comput and Appl Math 191:68–76

Halmos P (1985) I want to be a mathematician: An automathography. Springer

Higham N (1989) Matrix nearness problems and applications. In: Gover M, Barnett S (eds) Applications of Matrix Theory, Oxford University Press, pp 1–27

Hinrichsen D, Pritchard AJ (1986) Stability radius for structured perturbations and the algebraic Riccati equation. Control Lett 8:105–113

Jackson J (2003) A User's Guide to Principal Components. Wiley

Jolliffe I (2002) Principal component analysis. Springer-Verlag

Karmarkar N, Lakshman Y (1998) On approximate GCDs of univariate polynomials. In: Watt S, Stetter H (eds) J. Symbolic Computation, vol 26, pp 653–666

Kiers H (2002) Setting up alternating least squares and iterative majorization algorithms for solving various matrix optimization problems. Comput Statist Data Anal 41:157–170

Kim H, Park H (2007) Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis. Bioinformatics 23:1495–1502

Knuth D (1984) Literate programming. Comput J 27(2):97–111

Knuth D (1986) Computers & Typesetting, Volume B: TeX: The Program. Addison-Wesley

Knuth D (1992) Literate programming. Cambridge University Press

Koopmans T (1937) Linear regression analysis of economic time series. DeErven F. Bohn

Kovacevic J (2007) How to encourage and publish reproducible research. In: Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Proc., pp 1273–1276

Krim H, Viberg M (1996) Two decades of array signal processing research. IEEE Signal Processing Magazine 13:67–94

Kumaresan R, Tufts D (1983) Estimating the angles of arrival of multiple plane waves. IEEE Transactions on Aerospace and Electronic Systems 19(1):134–139

Ma Y, Soatto S, Kosecká J, Sastry S (2004) An Invitation to 3-D Vision, Interdisciplinary Applied Mathematics, vol 26. Springer

Madansky A (1959) The fitting of straight lines when both variables are subject to error. J Amer Statist Assoc 54:173–205

Pearson K (1901) On lines and planes of closest fit to points in space. Philos Mag 2:559–572

Polderman J, Willems JC (1998) Introduction to mathematical systems theory. Springer-Verlag, New York

Ramsey N (1994) Literate programming simplified. IEEE Software 11:97–105

Rump S (2003) Structured perturbations part I: Normwise distances. SIAM J Matrix Anal Appl 25:1–30

Schölkopf B, Smola A, Müller K (1999) Kernel principal component analysis., MIT Press, Cambridge, MA, pp 327–352

Shawe-Taylor J, Cristianini N (2004) Kernel Methods for Pattern Analysis. Cambridge University Press

Stewart GW (1993) On the early history of the singular value decomposition. SIAM Review 35(4):551–566

Tipping M, Bishop C (1999) Probabilistic principal component analysis. J R Stat Soc B 61(3):611–622

Tomasi C, Kanade T (1993) Shape and motion from image streames: A factorization method. Proc Natl Adadem Sci USA 90:9795–9802

Trefethen LN, Embree M (1999) Spectra and pseudospectra: The behavior of non-normal matrices and operators. Princeton university press

Vichia M, Saporta G (2009) Clustering and disjoint principal component analysis. Comput Statist Data Anal 53:3194–3208

Wentzell P, Andrews D, Hamilton D, Faber K, Kowalski B (1997) Maximum likelihood principal component analysis. J Chemometrics 11:339–366

Willems JC (1986, 1987) From time series to linear system—Part I. Finite dimensional linear time invariant systems, Part II. Exact modelling, Part III. Approximate modelling. Automatica 22, 23:561–580, 675–694, 87–115

Willems JC (1989) Models for dynamics. Dynamics reported 2:171–269

Willems JC (1991) Paradigms and puzzles in the theory of dynamical systems. IEEE Trans Automat Control 36(3):259–294