Chapter 2
From Data to Models

... whenever we have two different representations of the same thing we can learn a great deal by comparing representations and translating descriptions from one representation into the other. Shifting descriptions back and forth between representations can often lead to insights that are not inherent in either of the representations alone.

Abelson and diSessa (1986, p. 105)

2.1 Linear Static Model Representations

A linear static model with $\mathtt{q}$ variables is a subspace of $\mathbb{R}^{\mathtt{q}}$. We denote the set of linear static models with $\mathtt{q}$ variables by $\mathscr{L}_0^{\mathtt{q}}$. Three basic representations of a linear static model $\mathscr{B} \subseteq \mathbb{R}^{\mathtt{q}}$ are the kernel, image, and input/output ones:

- kernel representation

$$\mathscr{B} = \ker(R) := \big\{ d \in \mathbb{R}^{\mathtt{q}} \mid Rd = 0 \big\}, \qquad (\mathrm{KER}_0)$$

  with parameter $R \in \mathbb{R}^{\mathtt{p} \times \mathtt{q}}$,
- image representation

$$\mathscr{B} = \mathrm{image}(P) := \big\{ d = P\ell \in \mathbb{R}^{\mathtt{q}} \mid \ell \in \mathbb{R}^{\mathtt{m}} \big\}, \qquad (\mathrm{IMAGE}_0)$$

  with parameter $P \in \mathbb{R}^{\mathtt{q} \times \mathtt{m}}$, and
- input/output representation

$$\mathscr{B}_{\mathrm{i/o}}(X, \Pi) := \big\{ d = \Pi \operatorname{col}(u, y) \in \mathbb{R}^{\mathtt{q}} \mid u \in \mathbb{R}^{\mathtt{m}},\ y = X^\top u \big\}, \qquad (\mathrm{I/O}_0)$$

  with parameters $X \in \mathbb{R}^{\mathtt{m} \times \mathtt{p}}$ and a $\mathtt{q} \times \mathtt{q}$ permutation matrix Π.

If the parameter Π in an input/output representation is not specified, then by default it is assumed to be the identity matrix $\Pi = I_{\mathtt{q}}$, i.e., the first $\mathtt{m}$ variables are assumed inputs and the other $\mathtt{p} := \mathtt{q} - \mathtt{m}$ variables are outputs.

I. Markovsky, *Low Rank Approximation*, Communications and Control Engineering,
DOI 10.1007/978-1-4471-2227-2_2,

In the representation (IMAGE_0), the columns of P are *generators* of the model $\mathscr{B}$, i.e., they span or generate the model. In the representation (KER_0), the rows of R are *annihilators* of $\mathscr{B}$, i.e., they are orthogonal to the elements of $\mathscr{B}$. The parameters R and P are not unique, because

1. linearly dependent generators and annihilators can be added, respectively, to an existing set of annihilators and generators of $\mathscr{B}$ keeping the model the same, and

2. a set of generators or annihilators can be changed by an invertible transformation, without changing the model, i.e.,

$$\ker(R) = \ker(UR), \quad \text{for all } U \in \mathbb{R}^{\mathtt{p}\times\mathtt{p}}, \quad \text{such that} \quad \det(U) \neq 0;$$

and

$$\operatorname{image}(P) = \operatorname{image}(PV), \quad \text{for all } V \in \mathbb{R}^{\mathtt{m}\times\mathtt{m}}, \quad \text{such that} \quad \det(V) \neq 0.$$

The smallest possible number of generators, i.e., $\operatorname{col\,dim}(P)$, such that ($\text{IMAGE}_0$) holds is invariant of the representation and is equal to $\mathtt{m} := \dim(\mathscr{B})$—the dimension of $\mathscr{B}$. Integers, such as $\mathtt{m}$ and $\mathtt{p} := \mathtt{q} - \mathtt{m}$ that are invariant of the representation, characterize properties of the model (rather than of the representation) and are called *model invariants*. The integers $\mathtt{m}$ and $\mathtt{p}$ have data modeling interpretation as number of inputs and number of outputs, respectively. Indeed, $\mathtt{m}$ variables are free (unassigned by the model) and the other $\mathtt{p}$ variables are determined by the model and the inputs. The number of inputs and outputs of the model $\mathscr{B}$ are denoted by $\mathtt{m}(\mathscr{B})$ and $\mathtt{p}(\mathscr{B})$, respectively.

The model class of linear static models with $\mathtt{q}$ variables, at most $\mathtt{m}$ of which are inputs is denoted by $\mathscr{L}^{\mathtt{q}}_{\mathtt{m},0}$. With $\operatorname{col\,dim}(P) = \mathtt{m}$, the columns of P form a basis for $\mathscr{B}$. The smallest possible $\operatorname{row\,dim}(R)$, such that $\ker(R) = \mathscr{B}$ is invariant of the representation and is equal to the number of outputs of $\mathscr{B}$. With $\operatorname{row\,dim}(R) = \mathtt{p}$, the rows of R form a basis for the orthogonal complement $\mathscr{B}^{\perp}$ of $\mathscr{B}$. Therefore, without loss of generality we can assume that $P \in \mathbb{R}^{\mathtt{q}\times\mathtt{m}}$ and $R \in \mathbb{R}^{\mathtt{p}\times\mathtt{q}}$.

Exercise 2.1 Show that for $\mathscr{B} \in \mathscr{L}^{\mathtt{q}}_0$,

- $\min_{P,\operatorname{image}(P)=\mathscr{B}} \operatorname{col\,dim}(P)$ is equal to $\dim(\mathscr{B})$ and
- $\min_{R,\ker(R)=\mathscr{B}} \operatorname{row\,dim}(R)$ is equal to $\mathtt{q} - \dim(\mathscr{B})$.

In general, many input/output partitions of the variables d are possible. Choosing an input/output partition amounts to choosing a full rank $\mathtt{p} \times \mathtt{p}$ submatrix of R or a full rank $\mathtt{m} \times \mathtt{m}$ submatrix of P. In some data modeling problems, there is no a priori reason to prefer one partition of the variables over another. For such problems, the classical setting posing the problem as an overdetermined system of linear equations $AX \approx B$ is not a natural starting point.

Transition Among Input/Output, Kernel, and Image Representations

The transition from one model representation to another gives insight into the properties of the model. These *analysis problems* need to be solved before the more complicated modeling problems are considered. The latter can be viewed as *synthesis problems* since the model is created or synthesized from data and prior knowledge.

If the parameters R, P, and (X, Π) describe the same system $\mathscr{B}$, then they are related. We show the relations that the parameters must satisfy as well as code that does the transition from one representation to another. Before we describe the transition among the parameters, however, we need an efficient way to store and multiply by permutation matrices and a tolerance for computing rank numerically.

Input/Output Partition of the Variables

In the input/output model representation $\mathscr{B}_{\mathrm{i/o}}(X, \Pi)$, the partitioning of the variables $d \in \mathbb{R}^{\mathtt{q}}$ into inputs $u \in \mathbb{R}^{\mathtt{m}}$ and outputs $y \in \mathbb{R}^{\mathtt{p}}$ is specified by a permutation matrix Π,

$$d \underset{\Pi}{\overset{\Pi^{-1}=\Pi^\top}{\rightleftarrows}} \operatorname{col}(u, y), \qquad d = \Pi \begin{bmatrix} u \\ y \end{bmatrix}, \quad \begin{bmatrix} u \\ y \end{bmatrix} = \Pi^\top d.$$

In the software implementation, however, it is more convenient (as well as more memory and computation efficient) to specify the partitioning by a vector

$$\pi := \Pi \operatorname{col}(1, \ldots, \mathtt{q}) \in \{1, \ldots, \mathtt{q}\}^{\mathtt{q}}.$$

Clearly, the vector π contains the same information as the matrix Π and one can reconstruct Π from π by permuting the rows of the identity matrix. (In the code `io` is the variable corresponding to the vector π and `Pi` is the variable corresponding to the matrix Π.)

37a ⟨$\pi \mapsto \Pi$ 37a⟩≡ (38a)

```
Pi = eye(length(io)); Pi = Pi(io,:);
```

Permuting the elements of a vector is done more efficiently by direct reordering of the elements, instead of a matrix-vector multiplication. If `d` is a variable corresponding to a vector d, then `d(io)` corresponds to the vector Πd.

The default value for Π is the identity matrix I, corresponding to first `m` variables of d being inputs and the remaining `p` variables outputs, $d = \left[\begin{smallmatrix} u \\ y \end{smallmatrix}\right]$.

37b ⟨*default input/output partition* 37b⟩≡ (39 40a 42a)

```
if ~exist('io') || isempty(io), io = 1:q; end
```

In case the inverse permutation $\Pi^{-1} = \Pi^\top$ is needed, the corresponding "permutation vector" is

$$\pi' = \Pi^\top \operatorname{col}(1, \ldots, \mathtt{q}) \in \{1, \ldots, \mathtt{q}\}^{\mathtt{q}}.$$

In the code `inv_io` is the variable corresponding to the vector π' and the transition from the original variables `d` to the partitioned variables `uy = [u; y]` via `io` and `inv_io` is done by the following indexing operations:

$$\texttt{d} \underset{\texttt{io}}{\overset{\texttt{io\_inv}}{\rightleftarrows}} \texttt{uy}, \qquad \texttt{d = uy(io), uy = d(io\_inv).}$$

38a ⟨*inverse permutation* 38a⟩≡ (42b)

```
⟨π ↦ Π 37a⟩, inv_io = (1:length(io)) * Pi;
```

Tolerance for Rank Computation

In the computation of an input/output representation from a given kernel or image representation of a model (as well as in the computation of the models' complexity), we need to find the rank of a matrix. Numerically this is an ill-posed problem because arbitrary small perturbations of the matrix's elements may (generically will) change the rank. A solution to this problem is to replace rank with "numerical rank" defined as follows

$$\text{num rank}(A, \varepsilon) := \text{number of singular values of } A \text{ greater than } \varepsilon, \qquad \text{(num rank)}$$

where $\varepsilon \in \mathbb{R}_+$ is a user defined tolerance. Note that

$$\text{num rank}(A, 0) = \text{rank}(A),$$

i.e., by taking the tolerance to be equal to zero, the numerical rank reduces to the theoretical rank. A nonzero tolerance ε makes the numerical rank robust to perturbations of size (measured in the induced 2-norm) less than ε. Therefore, ε reflects the size of the expected errors in the matrix. The default tolerance is set to a small value, which corresponds to numerical errors due to a double precision arithmetic. (In the code `tol` is the variable corresponding to ε.)

38b ⟨*default tolerance* `tol` 38b⟩≡ (41–43 76b)

```
if ~exist('tol') || isempty(tol), tol = 1e-12; end
```

Note 2.2 The numerical rank definition (num rank) is the solution of an unstructured rank minimization problem: find a matrix $\widehat{A}$ of minimal rank, such that $\|A - \widehat{A}\|_2 < \varepsilon$.

From Input/Output Representation to Kernel or Image Representations

Consider a linear static model $\mathscr{B} \in \mathscr{L}^{\mathrm{q}}_{\mathrm{m},0}$, defined by an input/output representation $\mathscr{B}_{\mathrm{i/o}}(X, \Pi)$. From $y = X^\top u$, we have

$$\begin{bmatrix} X^\top & -I \end{bmatrix} \text{col}(u, y) = 0$$

or since $d = \Pi \left[\begin{smallmatrix} u \\ y \end{smallmatrix}\right]$,

$$\underbrace{\begin{bmatrix} X^\top & -I \end{bmatrix} \Pi^\top}_{R} \underbrace{\Pi \begin{bmatrix} u \\ y \end{bmatrix}}_{d} = 0.$$

Therefore, the matrix

$$R = \begin{bmatrix} X^\top & -I \end{bmatrix} \Pi^\top \qquad ((X, \Pi) \mapsto R)$$

is a parameter of a kernel representations of $\mathscr{B}$, i.e.,

$$\mathscr{B}_{\mathrm{i/o}}(X, \Pi) = \ker\left(\begin{bmatrix} X^\top & -I \end{bmatrix} \Pi^\top\right) = \mathscr{B}.$$

Moreover, the representation is minimal because R is full row rank.

Similarly, a minimal image representation is derived from the input/output representation as follows. From $y = X^\top u$,

$$\begin{bmatrix} u \\ y \end{bmatrix} = \begin{bmatrix} I \\ X^\top \end{bmatrix} u,$$

so that, using $d = \Pi \left[\begin{smallmatrix} u \\ y \end{smallmatrix}\right]$,

$$d = \Pi \begin{bmatrix} u \\ y \end{bmatrix} = \Pi \begin{bmatrix} I \\ X^\top \end{bmatrix} u =: Pu.$$

Therefore, the matrix

$$P = \Pi \begin{bmatrix} I \\ X^\top \end{bmatrix} \qquad ((X, \Pi) \mapsto P)$$

is a parameter of an image representations of $\mathscr{B}$, i.e.,

$$\mathscr{B}_{\mathrm{i/o}}(X, \Pi) = \mathrm{image}\left(\Pi \begin{bmatrix} I \\ X^\top \end{bmatrix}\right) = \mathscr{B}.$$

The representation is minimal because P is full column rank.

Formulas $((X, \Pi) \mapsto R)$ and $((X, \Pi) \mapsto P)$ give us a straight forward way of transforming a given input/output representation to minimal kernel and minimal image representations.

39 ⟨$(X, \Pi) \mapsto R$ 39⟩≡

```
function r = xio2r(x, io)
r = [x', -eye(size(x, 2))]; q = size(r, 2);
⟨default input/output partition 37b⟩, r = r(:, io);
```

Defines:
xio2r, used in chunk 45a.

40a ⟨$(X, \Pi) \mapsto P$ 40a⟩≡
```
function p = xio2p(x, io)
p = [eye(size(x, 1)); x']; q = size(p, 1);
⟨default input/output partition 37b⟩, p = p(io, :);
```
Defines:
xio2p, used in chunk 45a.

From Image to Minimal Kernel and from Kernel to Minimal Image Representation

The relation

$$\ker(R) = \text{image}(P) = \mathscr{B} \in \mathscr{L}^{\mathtt{q}}_{\mathtt{m},0} \quad \Longrightarrow \quad RP = 0 \qquad (R \leftrightarrow P)$$

gives a link between the parameters P and R. In particular, a minimal image representation $\text{image}(P) = \mathscr{B}$ can be obtained from a given kernel representation $\ker(R) = \mathscr{B}$ by computing a basis for the null space of R. Conversely, a minimal kernel representation $\ker(R) = \mathscr{B}$ can be obtained from a given image representation $\text{image}(P) = \mathscr{B}$ by computing a basis for the left null space of P.

40b ⟨$R \mapsto P$ 40b⟩≡
```
function p = r2p(r), p = null(r);
```
Defines:
r2p, used in chunks 44 and 45a.

40c ⟨$P \mapsto R$ 40c⟩≡
```
function r = p2r(p), r = null(p')';
```
Defines:
p2r, used in chunk 45a.

Converting an Image or Kernel Representation to a Minimal One

The kernel and image representations obtained from `xio2r`, `xio2p`, `p2r`, and `r2p` are minimal. In general, however, a given kernel or image representations can be non-minimal, i.e., R may have redundant rows and P may have redundant columns. The kernel representation, defined by R, is minimal if and only if R is full row rank. Similarly, the image representation, defined by P, is minimal if and only if P is full column rank.

The problems of converting kernel and image representations to minimal ones are equivalent to the problem of finding a full rank matrix that has the same kernel or image as a given matrix. A numerically reliable way to solve this problem is to use the singular value decomposition.

Consider a model $\mathscr{B} \in \mathscr{L}^{\mathtt{q}}_{\mathtt{m},0}$ with parameters $R \in \mathbb{R}^{g \times \mathtt{q}}$ and $P \in \mathbb{R}^{\mathtt{q} \times g}$ of, respectively, kernel and image representations. Let

$$R = U \Sigma V^{\top}$$

be the singular value decomposition of R and let $\mathtt{p}$ be the rank of R. With the partitioning,

$$V =: \begin{bmatrix} V_1 & V_2 \end{bmatrix}, \quad \text{where } V_1 \in \mathbb{R}^{\mathtt{q}\times\mathtt{p}},$$

we have

$$\text{image}\left(R^\top\right) = \text{image}(V_1).$$

Therefore,

$$\ker(R) = \ker\left(V_1^\top\right) \quad \text{and} \quad V_1 \text{ is full rank},$$

so that $V_1^\top$ is a parameter of a minimal kernel representation of $\mathscr{B}$.

Similarly, let

$$P = U\,\Sigma\,V^\top$$

be the singular value decomposition of P. With the partitioning,

$$U =: \begin{bmatrix} U_1 & U_2 \end{bmatrix}, \quad \text{where } U_1 \in \mathbb{R}^{\mathtt{q}\times\mathtt{m}},$$

we have

$$\text{image}(P) = \text{image}(U_1).$$

Since U_1 is full rank, it is a parameter of a minimal image representation of $\mathscr{B}$.

In the numerical implementation, the rank is replaced by the numerical rank with respect to a user defined tolerance `tol`.

41 ⟨$R \mapsto$ *minimal* R 41⟩≡

```
function r = minr(r, tol)
[p, q] = size(r); ⟨default tolerance tol 38b⟩
[u, s, v] = svd(r, 'econ'); pmin = sum(diag(s) > tol);
if pmin < p, r = v(:, 1:pmin)'; end
```

Defines:
`minr`, used in chunks 42b and 43c.

Exercise 2.3 Write a function `minp` that implements the transition $P \mapsto$ minimal P.

From Kernel or Image to Input/Output Representation

The transformations from kernel to input/output and from image to input/output representations are closely related. They involve as a sub-problem the problem of finding input/output partitions of the variables in the model. Because of this, they are more complicated than the inverse transformations, considered above.

Assume first that the input/output partition is given. This amounts to knowing the permutation matrix $\Pi \in \mathbb{R}^{\mathtt{q}\times\mathtt{q}}$ in (I/O$_0$).

42a ⟨$(R, \Pi) \mapsto X$ 42a⟩≡ 42b▷

```
function x = rio2x(r, io, tol)
q = size(r, 2); ⟨default input/output partition 37b⟩, ⟨default tolerance tol 38b⟩
```

Defines:
rio2x, used in chunks 43c and 45.

Consider given parameters $R \in \mathbb{R}^{\mathtt{p}\times\mathtt{q}}$ of *minimal* kernel representation of a linear static system $\mathscr{B} \in \mathscr{L}^{\mathtt{q}}_{\mathtt{m},0}$ and define the partitioning

$$R\Pi =: \begin{bmatrix} R_u & R_y \end{bmatrix}, \quad \text{where } R_y \in \mathbb{R}^{\mathtt{p}\times\mathtt{p}}.$$

42b ⟨$(R, \Pi) \mapsto X$ 42a⟩+≡ ◁42a 42c▷

```
r = minr(r, tol); p = size(r, 1); m = q - p;
⟨inverse permutation 38a⟩, rpi = r(:, inv_io);
ru  = rpi(:, 1:m); ry = rpi(:, (m + 1):end);
```

Uses minr 41.

Similarly, for a parameter $P \in \mathbb{R}^{\mathtt{q}\times\mathtt{m}}$ of *minimal* image representation of $\mathscr{B}$, define

$$\Pi^\top P =: \begin{bmatrix} P_u \\ P_y \end{bmatrix}, \quad \text{where } P_u \in \mathbb{R}^{\mathtt{m}\times\mathtt{m}}.$$

If R_y and P_u are non-singular, it follows from $((X, \Pi) \mapsto R)$ and $((X, \Pi) \mapsto P)$ that

$$X = -\left(R_y^{-1} R_u\right)^\top \qquad ((R, \Pi) \mapsto X)$$

and

$$X = \left(P_y P_u^{-1}\right)^\top \qquad ((P, \Pi) \mapsto X)$$

is the parameter of the input/output representation $\mathscr{B}_{\mathrm{i/o}}(X, \Pi)$ of $\mathscr{B}$, i.e.,

$$\ker\Bigg(\underbrace{\begin{matrix}\mathtt{m} & \mathtt{p}\\ \begin{bmatrix} R_u & R_y \end{bmatrix}\end{matrix}\Pi^\top}_{R}\Bigg) = \mathscr{B}_{\mathrm{i/o}}\left(-\left(R_y^{-1} R_u\right)^\top, \Pi\right)$$

and

$$\operatorname{image}\Bigg(\underbrace{\Pi \begin{bmatrix} P_u \\ P_y \end{bmatrix}}_{P} \begin{matrix}\mathtt{m}\\ \mathtt{p}\end{matrix}\Bigg) = \mathscr{B}_{\mathrm{i/o}}\left(\left(P_y P_u^{-1}\right)^\top, \Pi\right).$$

42c ⟨$(R, \Pi) \mapsto X$ 42a⟩+≡ ◁42b

```
[u, s, v] = svd(ry); s = diag(s);
if s(end) < tol
    warning('Computation of X is ill conditioned.');
    x = NaN;
else
    x = -( v * diag(1 ./ s) * u' * ru )';
end
```

Singularity of the blocks R_y and P_u implies that the input/output representation with a permutation matrix Π is not possible. In such cases, the function `rio2x` issues a warning message and returns `NaN` value for X.

The function `r2io` uses `rio2x` in order to find all possible input/output partitions for a model specified by a kernel representation.

43a ⟨$R \mapsto \Pi$ 43a⟩≡ 43b▷
```
function IO = r2io(r, tol)
q = size(r, 2); ⟨default tolerance tol 38b⟩
```
Defines:
r2io, used in chunk 43d.

The search is exhaustive over all input/output partitionings of the variables (i.e., all choices of `m` elements of the set of variable indices $\{1, \ldots, q\}$), so that the computation is feasible only for a small number of variables (say less than 6).

43b ⟨$R \mapsto \Pi$ 43a⟩+≡ ◁43a 43c▷
```
IO = perms(1:q); nio = size(IO, 1);
```

The parameter X for each candidate partition is computed. If the computation of X is ill conditioned, the corresponding partition is not consistent with the model and is discarded.

43c ⟨$R \mapsto \Pi$ 43a⟩+≡ ◁43b
```
not_possible = []; warning_state = warning('off');
r = minr(r, tol);
for i = 1:nio
    x = rio2x(r, IO(i, :), tol);
    if isnan(x), not_possible = [not_possible, i]; end
end
warning(warning_state); IO(not_possible, :) = [];
```
Uses minr 41 and rio2x 42a.

Example 2.4 Consider the linear static model with three variables and one input

$$\mathscr{B} = \text{image}\left(\begin{bmatrix}1\\0\\0\end{bmatrix}\right) = \ker\left(\begin{bmatrix}0 & 1 & 0\\0 & 0 & 1\end{bmatrix}\right).$$

Clearly, this model has only two input/output partitions:

$$u = w_1,\ y = \begin{bmatrix}w_2\\w_3\end{bmatrix} \quad \text{and} \quad u = w_1,\ y = \begin{bmatrix}w_3\\w_2\end{bmatrix}.$$

Indeed, the function `r2io`

43d ⟨*Test* r2io 43d⟩≡
```
r2io([0 0 1; 0 1 0])
```
Uses r2io 43a.

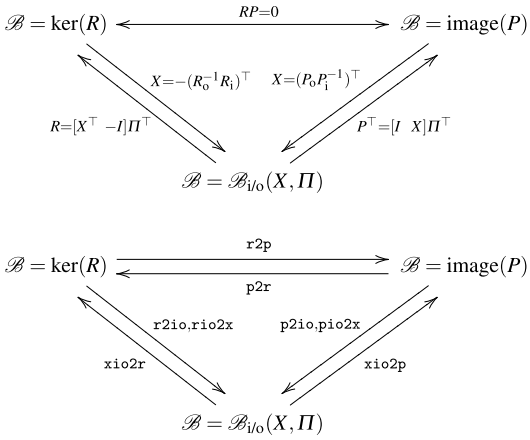


Fig. 2.1 Relations and functions for the transitions among linear static model parameters

correctly computes the input output partitionings from the parameter R in a kernel representation of the model

```
ans =

     1     2     3
     1     3     2
```

Exercise 2.5 Write functions `pio2x` and `p2io` that implement the transitions

$$(P, \Pi) \mapsto X \quad \text{and} \quad P \mapsto \Pi.$$

Exercise 2.6 Explain how to check that two models, specified by kernel or image representation, are equivalent.

Summary of Transitions among Representations

Figure 2.1 summarizes the links among the parameters R, P, and (X, Π) of a linear static model $\mathscr{B}$ and the functions that implement the transitions.

Numerical Example

In order to test the functions for transition among the kernel, image, and input/output model representations, we choose, a random linear static model, specified by a kernel representation,

44 ⟨*Test model transitions* 44⟩≡ 45a▷

```
m = 2; p = 3; R = rand(p, m + p); P = r2p(R);
```

Uses `r2p` 40b.

and traverse the diagram in Fig. 2.1 clock-wise and anti clock-wise

45a ⟨*Test model transitions* 44⟩+≡ ◁44 45b▷
```
R_ = xio2r(pio2x(r2p(R))); P_ = xio2p(rio2x(p2r(P)));
```
Uses p2r 40c, r2p 40b, rio2x 42a, xio2p 40a, and xio2r 39.

As a verification of the software, we check that after traversing the loop, equivalent models are obtained.

45b ⟨*Test model transitions* 44⟩+≡ ◁45a
```
norm(rio2x(R) - rio2x(R_)), norm(pio2x(P) - pio2x(P_))
```
Uses rio2x 42a.

The answers are around the machine precision, which confirms that the models are the same.

Linear Static Model Complexity

A linear static model is a finite dimensional subspace. The dimension m of the subspace is equal to the number of inputs and is invariant of the model representation. The integer constant m quantifies the *model complexity*: the model is more complex when it has more inputs. The rationale for this definition of model complexity is that inputs are "unexplained" variables by the model, so the more inputs the model has, the less it "explains" the modeled phenomenon. In data modeling the aim is to obtain low-complexity models, a principle generally referred to as *Occam's razor*.

Note 2.7 (Computing the model complexity is a rank estimation problem) Computing the model complexity, implicitly specified by (exact) data, or by a non-minimal kernel or image representation is a rank computation problem; see Sect. 2.3 and the function minr.

2.2 Linear Time-Invariant Model Representations

An observation d_j of a static model is a vector of variables. In the dynamic case, the observations depend on time, so that apart from the multivariable aspect, there is also a time evaluation aspect. In the dynamic case, an observation is referred to as a *trajectory*, i.e., it is a vector valued *function* of a scalar argument. The time variable takes its values in the set of integers $\mathbb{Z}$ (*discrete-time* model) or in the set of real numbers $\mathbb{R}$ (*continuous-time* model). We denote the time axis by $\mathscr{T}$.

A dynamic model $\mathscr{B}$ with q variables is a subset of the trajectory space $(\mathbb{R}^q)^{\mathscr{T}}$—the set of all functions from the time axis $\mathscr{T}$ to the variable space $\mathbb{R}^q$.

In this book, we consider the special class of finite dimensional linear time-invariant dynamical models. By definition, a model $\mathscr{B}$ is *linear* if it is a subspace of the data space $(\mathbb{R}^{\mathtt{q}})^{\mathscr{T}}$. In order to define the *time-invariance* property, we introduce the *shift operator* σ^τ. Acting on a signal w, σ^τ produces a signal $\sigma^\tau w$, which is the backwards shifted version of w by τ time units, i.e.,

$$\big(\sigma^\tau w\big)(t) := w(t+\tau), \quad \text{for all } t \in \mathscr{T}.$$

Acting on a set of trajectories, σ^τ shifts all trajectories in the set, i.e.,

$$\sigma^\tau \mathscr{B} := \big\{\sigma^\tau w \mid w \in \mathscr{B}\big\}.$$

A model $\mathscr{B}$ is shift-invariant if it is invariant under any shift in time, i.e.,

$$\sigma^\tau \mathscr{B} = \mathscr{B}, \quad \text{for all } \tau \in \mathscr{T}.$$

The model $\mathscr{B}$ is *finite dimensional* if it is a closed subset (in the topology of pointwise convergence). Finite dimensionality is equivalent to the property that at any time t the future behavior of the model is deterministically independent of the past behavior, given a finite dimensional vector, called a *state* of the model. Intuitively, the state is the information (or memory) of the past that is needed in order to predict the future. The smallest state dimension is an invariant of the system, called the *order*. We denote the set of finite dimensional linear time-invariant models with $\mathtt{q}$ variables and order at most $\mathtt{n}$ by $\mathscr{L}^{\mathtt{q},\mathtt{n}}$ and the *order* of $\mathscr{B}$ by $\mathtt{n}(\mathscr{B})$.

A finite dimensional linear time-invariant model $\mathscr{B} \in \mathscr{L}^{\mathtt{q},\mathtt{n}}$ admits a representation by a difference or differential equation

$$\begin{aligned} R_0 w + R_1 \lambda w + \cdots + R_{\mathtt{l}} \lambda^{\mathtt{l}} w &= \underbrace{\big(R_0 + R_1 \lambda + \cdots + R_{\mathtt{l}} \lambda^{\mathtt{l}}\big)}_{R(\lambda)} w \\ &= R(\lambda) w = 0, \end{aligned} \tag{DE}$$

where λ is the unit shift operator σ in the discrete-time case and the differential operator $\frac{\mathrm{d}}{\mathrm{d}t}$ in the continuous-time case. Therefore, the model $\mathscr{B}$ is the kernel

$$\mathscr{B} := \ker\big(R(\lambda)\big) = \big\{w \mid (\text{DE}) \text{ holds}\big\}, \tag{KER}$$

of the difference or differential operator $R(\lambda)$. The smallest degree $\mathtt{l}$ of a polynomial matrix

$$R(z) := R_0 + R_1 z + \cdots + R_{\mathtt{l}} z^{\mathtt{l}} \in \mathbb{R}^{g \times \mathtt{q}}[z],$$

in a kernel representation (KER) of $\mathscr{B}$, is invariant of the representation and is called the *lag* $\mathtt{l}(\mathscr{B})$ of $\mathscr{B}$.

The order of the system is the total degree of the polynomial matrix R in a kernel representation of the system. Therefore, we have the following link between the order and the lag of a linear time-invariant model:

$$\mathtt{n}(\mathscr{B}) \leq \mathtt{p}(\mathscr{B})\mathtt{l}(\mathscr{B}).$$

As in the static case, the smallest possible number of rows g of the polynomial matrix R in a kernel representation (KER) of a finite dimensional linear time-invariant system $\mathscr{B}$ is the invariant $\mathtt{p}(\mathscr{B})$—number of outputs of $\mathscr{B}$. Finding an input/output partitioning for a model specified by a kernel representation amounts to selection of a non-singular $\mathtt{p} \times \mathtt{p}$ submatrix of R. The resulting input/output representation is:

$$\mathscr{B} = \mathscr{B}_{\mathrm{i/o}}(P, Q, \Pi) := \ker\big(\Pi \begin{bmatrix} Q(\lambda) & P(\lambda) \end{bmatrix}\big), \tag{I/O}$$

with parameters the polynomial matrices

$$Q \in \mathbb{R}^{\mathtt{p}\times\mathtt{m}}[z] \quad \text{and} \quad P \in \mathbb{R}^{\mathtt{p}\times\mathtt{p}}[z], \quad \text{such that} \quad \det(P) \neq 0,$$

and the permutation matrix Π.

In general, the representation (I/O) involves higher order shifts or derivatives. A first order representation

$$\mathscr{B} = \mathscr{B}_{\mathrm{i/s/o}}(A, B, C, D, \Pi) := \big\{ w = \Pi \operatorname{col}(u, y) \mid \text{there is } x, \text{ such that } \lambda x = Ax + Bu \text{ and } y = Cx + Du \big\}, \tag{I/S/O}$$

with an auxiliary variable x, however, is always possible. The representation (I/S/O) displays not only the input/output structure of the model but also its state structure and is referred to as an input/state/output representation of the model. The parameters of an input/state/output representation are the matrices

$$A \in \mathbb{R}^{\mathtt{n}\times\mathtt{n}}, \qquad B \in \mathbb{R}^{\mathtt{n}\times\mathtt{m}}, \qquad C \in \mathbb{R}^{\mathtt{p}\times\mathtt{n}}, \qquad D \in \mathbb{R}^{\mathtt{p}\times\mathtt{m}},$$

and a permutation matrix Π. The parameters are nonunique due to

- nonuniqueness in the choice of the input/output partition,
- existence of redundant states (non-minimality of the representation), and
- change of state space basis

$$\mathscr{B}_{\mathrm{i/s/o}}(A, B, C, D) = \mathscr{B}_{\mathrm{i/s/o}}\big(T^{-1}AT, T^{-1}B, CT, D\big), \quad \text{for any non-singular matrix } T \in \mathbb{R}^{\mathtt{n}\times\mathtt{n}}. \tag{CB}$$

An input/state/output representation $\mathscr{B}_{\mathrm{i/s/o}}(A, B, C, D)$ is called *minimal* when the state dimension $\mathtt{n}$ is as small as possible. The minimal state dimension is an invariant of the system and is equal to the order $\mathtt{n}(\mathscr{B})$ of the system.

A system $\mathscr{B}$ is *autonomous* if for any trajectory $w \in \mathscr{B}$ the "past"

$$w_- := \big(\ldots, w(-2), w(-1)\big)$$

of w completely determines its "future"

$$w_+ := \big(w(0), w(1), \ldots\big).$$

Table 2.1 Summary of model $\mathscr{B} \subset (\mathbb{R}^{\mathtt{q}})^{\mathscr{T}}$ properties

Property	Definition
linearity	$w, v \in \mathscr{B} \implies \alpha w + \beta v \in \mathscr{B}$, for all $\alpha, \beta \in \mathbb{R}$
time-invariance	$\sigma^{\tau}\mathscr{B} = \mathscr{B}$, for all $\tau \in \mathscr{T}$
finite dimensionality	$\mathscr{B}$ is a closed set; equivalently $\mathtt{n}(\mathscr{B}) < \infty$
autonomy	the past of any trajectory completely determines its future; equivalently $\mathtt{m}(\mathscr{B}) = 0$
controllability	the past of any trajectory can be concatenated to the future of any other trajectory by a third trajectory if a transition period is allowed

It can be shown that a system $\mathscr{B}$ is autonomous if and only if it has no inputs. An autonomous finite dimensional linear time-invariant system is parametrized by the pair of matrices A and C via the state space representation

$$\mathscr{B}_{\mathrm{i/s/o}}(A, C) := \big\{ w = y \mid \text{there is } x, \text{ such that } \sigma x = Ax \text{ and } y = Cx \big\}.$$

The dimension $\dim(\mathscr{B})$ of an autonomous linear model $\mathscr{B}$ is equal to the order $\mathtt{n}(\mathscr{B})$.

In a way the opposite of an autonomous model is a controllable system. The model $\mathscr{B}$ is *controllable* if for any trajectories w_{p} and w_{f} of $\mathscr{B}$, there is $\tau > 0$ and a third trajectory $w \in \mathscr{B}$, which coincides with w_{p} in the past, i.e., $w(t) = w_{\mathrm{p}}(t)$, for all $t < 0$, and coincides with w_{f} in the future, i.e., $w(t) = w_{\mathrm{f}}(t)$, for all $t \geq \tau$. The subset of controllable systems of the set of linear time-invariant systems $\mathscr{L}^{\mathtt{q}}$ is denoted by $\mathscr{L}^{\mathtt{q}}_{\mathrm{ctrb}}$. A summary of properties of a dynamical system is given in Table 2.1.

Apart from the kernel, input/output, and input/state/output representation, a controllable finite dimensional linear time-invariant model admits the following representations:

- *image representation*

$$\mathscr{B} = \mathrm{image}\big(P(\lambda)\big) := \big\{ w \mid w = P(\lambda)\ell, \text{ for some } \ell \big\}, \tag{IMAGE}$$

  with parameter the polynomial matrix $P(z) \in \mathbb{R}^{\mathtt{q} \times g}[z]$,
- *convolution representation*

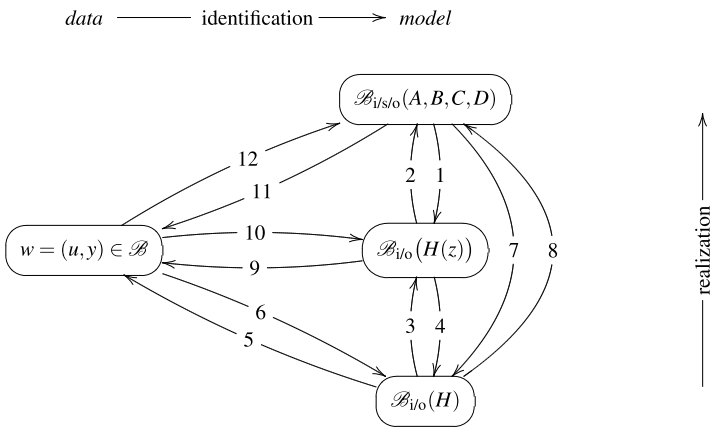
$$\mathscr{B} = \mathscr{B}_{\mathrm{i/o}}(H, \Pi) := \big\{ w = \Pi \,\mathrm{col}(u, y) \mid y = H \star u \big\}, \tag{CONV}$$

  where $\star$ is the convolution operator

$$y(t) = (H \star u)(t) := \sum_{\tau=0}^{\infty} H(\tau) u(t - \tau), \quad \text{in discrete-time, or}$$

$$y(t) = (H \star u)(t) := \int_0^{\infty} H(\tau) u(t - \tau)\, \mathrm{d}\tau, \quad \text{in continuous-time,}$$

  with parameters the signal $H : \mathscr{T} \to \mathbb{R}^{\mathtt{p} \times \mathtt{m}}$ and a permutation matrix Π; and



1. $H(z) = C(Iz - A)^{-1}B + D$
2. realization of a transfer function
3. Z or Laplace transform of $H(t)$
4. inverse transform of $H(z)$
5. convolution $y_{\mathrm{d}} = H \star u_{\mathrm{d}}$
6. exact identification
7. $H(0) = D$, $H(t) = CA^{t-1}B$ (discrete-time), $H(t) = Ce^{At}B$ (continuous-time), for $t > 0$
8. realization of an impulse response
9. simulation with input u_{d} and $x(0) = 0$
10. exact identification
11. simulation with input u_{d} and $x(0) = x_{\mathrm{ini}}$
12. exact identification

Fig. 2.2 Data, input/output model representations, and links among them

- *transfer function*,

$$\mathscr{B} = \mathscr{B}_{\mathrm{i/o}}(H, \Pi) := \left\{ w = \Pi \operatorname{col}(u,\, y) \mid \mathscr{F}(y) = H(z)\mathscr{F}(u) \right\}, \qquad \text{(TF)}$$

  where $\mathscr{F}$ is the Z-transform in discrete-time and the Laplace transform in continuous-time, with parameters the rational matrix $H \in \mathbb{R}^{\mathtt{p}\times\mathtt{m}}(s)$ (the transfer function) and a permutation matrix Π.

Transitions among the parameters H, $H(z)$, and (A, B, C, D) are classical problems, see Fig. 2.2. Next, we review the transition from impulse response H to parameters (A, B, C, D) of an input/state/output representation, which plays an important role in deriving methods for Hankel structured low rank approximation.

System Realization

The problem of passing from a convolution representation to an input/state/output representation is called (impulse response) realization.

Definition 2.8 (Realization) A linear time-invariant system $\mathscr{B}$ with $\mathtt{m}$ inputs and $\mathtt{p}$ outputs and an input/output partition, specified by a permutation matrix Π, is a *realization* of (or realizes) an impulse response $H : \mathscr{T} \to \mathbb{R}^{\mathtt{p}\times\mathtt{m}}$ if $\mathscr{B}$ has a convolution

representation $\mathscr{B} = \mathscr{B}_{\text{i/o}}(H, \Pi)$. A realization $\mathscr{B}$ of H is *minimal* if its order $\mathtt{n}(\mathscr{B})$ is the smallest over all realization of H.

In what follows, we fix the input/output partition Π to the default one

$$w = \text{col}(u, y)$$

and use the notation

$$H := \begin{bmatrix} h_1 \cdots h_{\mathtt{m}} \end{bmatrix} \quad \text{and} \quad I_{\mathtt{m}} := \begin{bmatrix} e_1 \cdots e_{\mathtt{m}} \end{bmatrix}.$$

An equivalent definition of impulse response realization that makes explicit the link of the realization problem to data modeling is the following one.

Definition 2.9 (Realization, as a data modeling problem) The system $\mathscr{B}$ realizes H if the set of input/output trajectories (e_i, h_i), for $i = 1, \ldots, \mathtt{m}$ are impulse responses of $\mathscr{B}$, i.e.,

$$(e_i \delta, 0 \wedge h_i) \in \mathscr{B}, \quad \text{for } i = 1, \ldots, \mathtt{m},$$

where δ is the delta function and $\wedge$ is the concatenation map (at time 0)

$$w = w_{\text{p}} \wedge w_{\text{f}}, \qquad w(t) := \begin{cases} w_{\text{p}}(t), & \text{if } t < 0 \\ w_{\text{f}}(t), & \text{if } t \geq 0 \end{cases}$$

Note 2.10 (Discrete-time vs. continuous-time system realization) There are some differences between the discrete and continuous-time realization theory. Next, we consider the discrete-time case. It turns out, however, that the discrete-time algorithms can be used for realization of continuous-time systems by applying them on the sequence of the Markov parameter $(H(0), \frac{\text{d}}{\text{d}t} H(0), \ldots)$ of the system.

The sequence

$$H = \big(H(0), H(1), H(2), \ldots, H(t), \ldots\big), \quad \text{where } H(t) \in \mathbb{R}^{\mathtt{p} \times \mathtt{m}}$$

is a one sided infinite matrix-values time series. Acting on H, the shift operator σ, removes the first sample, i.e.,

$$\sigma H = \big(H(1), H(2), \ldots, H(t), \ldots\big).$$

A sequence H might not be realizable by a *finite dimensional* linear time-invariant system, but if it is realizable, a minimal realization is unique.

Theorem 2.11 (Test for realizability) *The sequence $H : \mathbb{Z}_+ \to \mathbb{R}^{\mathtt{p} \times \mathtt{m}}$ is realizable by a finite dimensional linear time-invariant system with $\mathtt{m}$ inputs if and only if the two-sided infinite Hankel matrix $\mathscr{H}(\sigma H)$ has a finite rank $\mathtt{n}$. Moreover, the order of a minimal realization is equal to $\mathtt{n}$, and there is a unique system $\mathscr{B}$ in $\mathscr{L}_{\mathtt{m}}^{\mathtt{q},\mathtt{n}}$ that realizes H.*

Proof ($\Longrightarrow$) Let H be realizable by a system $\mathscr{B} \in \mathscr{L}_{\mathtt{m}}^{\mathtt{q},\mathtt{n}}$ with a minimal input/state/output representation $\mathscr{B} = \mathscr{B}_{\mathrm{i/s/o}}(A, B, C, D)$. Then

$$H(0) = D \quad \text{and} \quad H(t) = CA^{t-1}B, \quad \text{for } t > 0.$$

The (i, j) block element of the Hankel matrix $\mathscr{H}(\sigma H)$ is

$$H(i + j - 1) = CA^{i+j-2}B = CA^{i-1}A^{j-1}B.$$

Let

$$\mathscr{O}_t(A, C) := \mathrm{col}\big(C, CA, \ldots, CA^{t-1}\big) \tag{$\mathscr{O}$}$$

be the extended observability matrix of the pair (A, C) and

$$\mathscr{C}_t(A, B) := \big[B \; AB \cdots A^{t-1}B\big] \tag{$\mathscr{C}$}$$

be the extended controllability matrix of the pair (A, B). With $\mathscr{O}(A, C)$ and $\mathscr{C}(A, B)$ being the infinite observability and controllability matrices, we have

$$\mathscr{H}(\sigma H) = \mathscr{O}(A, C)\mathscr{C}(A, B) \tag{$\mathscr{O}\mathscr{C}$}$$

Since the representation $\mathscr{B}_{\mathrm{i/s/o}}(A, B, C, D)$ is assumed to be minimal, $\mathscr{C}(A, B)$ is full row rank and $\mathscr{O}(A, C)$ is full column rank. Therefore, ($\mathscr{O}\mathscr{C}$) is a rank revealing factorization of $\mathscr{H}(\sigma H)$ and

$$\mathrm{rank}\big(\mathscr{H}(\sigma H)\big) = \mathtt{n}(\mathscr{B}).$$

($\Longleftarrow$) In this direction, the proof is constructive and results in an *algorithm* for computation of the minimal realization of H in $\mathscr{L}_{\mathtt{m}}^{\mathtt{q},\mathtt{n}}$, where $\mathtt{n} = \mathrm{rank}(\mathscr{H}(\sigma H))$. A realization algorithm is presented in Sect. 3.1. □

Theorem 2.11 shows that

$$\mathrm{rank}\big(\mathscr{H}_{i,j}(\sigma H)\big) = \mathtt{n}(\mathscr{B}), \quad \text{for } \mathtt{p}i \geq \mathtt{n}(\mathscr{B}) \text{ and } \mathtt{m}j \geq \mathtt{n}(\mathscr{B}).$$

This suggests a method to find the order $\mathtt{n}(\mathscr{B})$ of the minimal realization of H: compute the rank of the *finite* Hankel matrix $\mathscr{H}_{i,j}(\sigma H)$, where $\mathtt{n}_{\max} := \min(\mathtt{p}i, \mathtt{m}j)$ is an upper bound of the order. Algorithms for computing the order and parameters of the minimal realization are presented in Chap. 3.1.

Linear Time-Invariant Model Complexity

Associate with a linear time-invariant dynamical system $\mathscr{B}$, we have defined the following system invariants:

$$\begin{array}{lll} \mathtt{m}(\mathscr{B}) \text{ number of inputs}, & & \mathtt{n}(\mathscr{B}) \text{ order}, \\ \mathtt{p}(\mathscr{B}) \text{ number of outputs} & \text{and} & \mathtt{l}(\mathscr{B}) \text{ lag}. \end{array}$$

The complexity of a linear static model $\mathscr{B}$ is the number of inputs $\mathtt{m}(\mathscr{B})$ of $\mathscr{B}$ or, equivalently, the dimension $\dim(\mathscr{B})$ of $\mathscr{B}$. Except for the class of autonomous systems, however, the dimension of a dynamical model is infinite. We define the restriction $\mathscr{B}|_{[1,T]}$ of $\mathscr{B}$ to the interval $[1, T]$,

$$\mathscr{B}|_{[1,T]} := \left\{ w \in \mathbb{R}^T \mid \text{there exist } w_{\mathrm{p}} \text{ and } w_{\mathrm{f}}, \text{ such that } (w_{\mathrm{p}}, w, w_{\mathrm{f}}) \in \mathscr{B} \right\}. \tag{$\mathscr{B}|_{[1,T]}$}$$

For a linear time-invariant model $\mathscr{B}$ and for $T > \mathtt{n}(\mathscr{B})$,

$$\dim(\mathscr{B}|_{[1,T]}) = \mathtt{m}(\mathscr{B})T + \mathtt{n}(\mathscr{B}) \leq \mathtt{m}(\mathscr{B})T + \mathtt{l}(\mathscr{B})\mathtt{p}(\mathscr{B}), \tag{dim $\mathscr{B}$}$$

which shows that the pairs of natural numbers

$$\big(\mathtt{m}(\mathscr{B}), \mathtt{n}(\mathscr{B})\big) \quad \text{and} \quad \big(\mathtt{m}(\mathscr{B}), \mathtt{l}(\mathscr{B})\big)$$

characterize the model's complexity. The elements of the model class $\mathscr{L}^{\mathtt{q}}_{\mathtt{m},\mathtt{l}}$ are linear time-invariant systems of complexity bounded by the pair $(\mathtt{m}, \mathtt{l})$ and, similarly, the elements of the model class $\mathscr{L}^{\mathtt{q},\mathtt{n}}_{\mathtt{m}}$ are linear time-invariant systems of complexity bounded by the pair $(\mathtt{m}, \mathtt{n})$. A static model is a special case of a dynamic model when the lag (or the order) is zero. This is reflected in the notation $\mathscr{L}^{\mathtt{q},\mathtt{n}}_{\mathtt{m},\mathtt{l}}$: the linear static model class $\mathscr{L}_{\mathtt{m},0}$ corresponds to the linear time-invariant model class $\mathscr{L}_{\mathtt{m},\mathtt{l}}$ with $\mathtt{l} = 0$.

Note that in the autonomous case, i.e., with $\mathtt{m}(\mathscr{B}) = 0$, $\dim(\mathscr{B}) = \mathtt{n}$. The dimension of the system corresponds to the number of degrees of freedom in selecting a trajectory. In the case of an autonomous system, the trajectory depends only on the initial condition (an $\mathtt{n}(\mathscr{B})$ dimensional vector). In the presence of inputs, the number of degrees of freedom due to the initial condition is increased on each time step by the number of inputs, due to the free variables. Asymptotically as $T \to \infty$, the term $\mathtt{m}T$ in (dim $\mathscr{B}$) dominates the term $\mathtt{n}$. Therefore, in comparing linear time-invariant system's complexities, by convention, a system with more inputs is more complex than a system with less inputs, irrespective of their state dimensions.

2.3 Exact and Approximate Data Modeling

General Setting for Data Modeling

In order to treat static, dynamic, linear, and nonlinear modeling problems with unified terminology and notation, we need an abstract setting that is general enough to accommodate all envisaged applications. Such a setting is described in this section.

The data $\mathscr{D}$ and a model $\mathscr{B}$ for the data are subsets of a *universal set* $\mathscr{U}$ of possible observations. In static modeling problems, $\mathscr{U}$ is a real $\mathtt{q}$-dimensional vector space $\mathbb{R}^{\mathtt{q}}$, i.e., the observations are real valued vectors. In dynamic modeling problems, $\mathscr{U}$ is a function space $(\mathbb{R}^{\mathtt{q}})^{\mathscr{T}}$, with $\mathscr{T}$ being $\mathbb{Z}$ in the discrete-time case and $\mathbb{R}$ in the continuous-time case.

Note 2.12 (Categorical data and finite automata) In modeling problems with categorical data and finite automata, the universal set $\mathscr{U}$ is discrete and may be finite.

We consider data sets $\mathscr{D}$ consisting of a finite number of observations

$$\mathscr{D} = \{w_{\mathrm{d},1}, \ldots, w_{\mathrm{d},N}\} \subset \mathscr{U}.$$

In discrete-time dynamic modeling problems, the $w_{\mathrm{d},j}$'s are trajectories, i.e.,

$$w_{\mathrm{d},j} = \big(w_{\mathrm{d},j}(1), \ldots, w_{\mathrm{d},j}(T_j)\big), \quad \text{with } w_{\mathrm{d},j}(t) \in \mathbb{R}^{\mathtt{q}} \text{ for all } t.$$

In dynamic problems, the data $\mathscr{D}$ often consists of a single trajectory $w_{\mathrm{d},1}$, in which case the subscript index 1 is skipped and $\mathscr{D}$ is identified with w_{d}. In static modeling problems, an observation $w_{\mathrm{d},j}$ is a vector and the alternative notation $d_j = w_{\mathrm{d},j}$ is used in order to emphasize the fact that the observations do not depend on time.

Note 2.13 (Given data vs. general trajectory) In order to distinguish a general trajectory w of the system from the given data w_{d} (a specific trajectory) we use the subscript "d" in the notation of the given data.

A *model class* $\mathscr{M}$ is a set of sets of $\mathscr{U}$, i.e., $\mathscr{M}$ is an element of the power set $2^{\mathscr{U}}$ of $\mathscr{U}$. We consider the generic model classes of

- linear static models $\mathscr{L}_0$,
- linear time-invariant models $\mathscr{L}$, and
- polynomial static models $\mathscr{P}$ (see Chap. 6).

In some cases, however, subclasses of the generic classes above are of interest. For examples, the controllable and finite impulse response model subclasses of the class of linear time-invariant models, and the ellipsoids subclass of the class of second order polynomial models (conic sections).

The complexity $\mathtt{c}$ of a model $\mathscr{B}$ is a vector of positive integers

$$\mathtt{c}(\mathscr{B}) := \begin{cases} \mathtt{m}(\mathscr{B}) = \dim(\mathscr{B}), & \text{if } \mathscr{B} \in \mathscr{L}_0, \\ (\mathtt{m}(\mathscr{B}), \mathtt{l}(\mathscr{B})) \text{ or } (\mathtt{m}(\mathscr{B}), \mathtt{n}(\mathscr{B})), & \text{if } \mathscr{B} \in \mathscr{L}, \\ (\mathtt{m}(\mathscr{B}), \deg(R)), \text{ where } \mathscr{B} = \ker(R), & \text{if } \mathscr{B} \in \mathscr{P}. \end{cases} \qquad (\mathtt{c}(\mathscr{B}))$$

Complexities are compared in this book by the lexicographic ordering, i.e., two complexities are compared by comparing their corresponding elements in the increasing order of the indices. The first time an index is larger, the corresponding complexity is declared larger. For linear time-invariant dynamic models, this convention and the ordering of the elements in $\mathtt{c}(\mathscr{B})$ imply that a model with more inputs is always more complex than a model with less inputs irrespective of their orders.

The complexity $\mathtt{c}$ of a model class $\mathscr{M}$ is the largest complexity of a model in the model class. Of interest is the restriction of the generic model classes $\mathscr{M}$ to subclasses $\mathscr{M}_{\mathtt{c}_{\max}}$ of models with bounded complexity, e.g., $\mathscr{L}^{\mathtt{q}}_{\mathtt{m},\mathtt{l}_{\max}}$, with $\mathtt{m} < \mathtt{q}$.

Exact Data Modeling

A model $\mathscr{B}$ is an *exact model* for the data $\mathscr{D}$ if $\mathscr{D} \subset \mathscr{B}$. Otherwise, it is an *approximate model*. An exact model for the data may not exist in a model class $\mathscr{M}_{\mathsf{C}_{\max}}$ of bounded complexity. This is generically the case when the data are noisy and the data set $\mathscr{D}$ is large enough (relative to the model complexity). A practical data modeling problem must involve approximation. Our starting point, however, is the simpler problem of exact data modeling.

Problem 2.14 (Exact data modeling) Given data $\mathscr{D} \subset \mathscr{U}$ and a model class $\mathscr{M}_{\mathsf{C}_{\max}} \in 2^{\mathscr{U}}$, find a model $\widehat{\mathscr{B}}$ in $\mathscr{M}_{\mathsf{C}_{\max}}$ that contains the data and has minimal (in the lexicographic ordering) complexity or assert that such a model does not exist, i.e.,

$$\text{minimize} \quad \text{over } \mathscr{B} \in \mathscr{M}_{\mathsf{C}_{\max}} \quad \mathsf{C}(\mathscr{B}) \quad \text{subject to} \quad \mathscr{D} \subset \mathscr{B} \qquad \text{(EM)}$$

The question occurs:

> (Existence of exact model) Under what conditions on the data $\mathscr{D}$ and the model class $\mathscr{M}_{\mathsf{C}_{\max}}$ does a solution to problem (EM) exist?

If a solution exists, it is unique. This unique solution is called the *most powerful unfalsified model* for the data $\mathscr{D}$ in the model class $\mathscr{M}_{\mathsf{C}_{\max}}$ and is denoted by $\mathscr{B}_{\mathrm{mpum}}(\mathscr{D})$. (The model class $\mathscr{M}_{\mathsf{C}_{\max}}$ is not a part of the notation $\mathscr{B}_{\mathrm{mpum}}(\mathscr{D})$ and is understood from the context.)

Suppose that the data $\mathscr{D}$ are generated by a model $\mathscr{B}_0$ in the model class $\mathscr{M}_{\mathsf{C}_{\max}}$, i.e.,

$$\mathscr{D} \subset \mathscr{B}_0 \in \mathscr{M}_{\mathsf{C}_{\max}}.$$

Then, the exact modeling problem has a solution in the model class $\mathscr{M}_{\mathsf{C}_{\max}}$, however, the solution $\mathscr{B}_{\mathrm{mpum}}(\mathscr{D})$ may not be the data generating model $\mathscr{B}_0$. The question occurs:

> (Identifiability) Under what conditions on the data $\mathscr{D}$, the data generating model $\mathscr{B}_0$, and the model class $\mathscr{M}_{\mathsf{C}_{\max}}$, the most powerful unfalsified model $\mathscr{B}_{\mathrm{mpum}}(\mathscr{D})$ in $\mathscr{M}_{\mathsf{C}_{\max}}$ coincides with the data generating model $\mathscr{B}_0$?

Example 2.15 (Exact data fitting by a linear static model) Existence of a linear static model $\widehat{\mathscr{B}}$ of bounded complexity $\mathtt{m}$ for the data $\mathscr{D}$ is equivalent to rank deficiency of the matrix

$$\Phi(\mathscr{D}) := \begin{bmatrix} d_1 & \cdots & d_N \end{bmatrix} \in \mathbb{R}^{\mathtt{q} \times N},$$

composed of the data. (Show this.) Moreover, the rank of the matrix $\Phi(\mathscr{D})$ is equal to the *minimal* dimension of an exact model for $\mathscr{D}$

$$\text{existence of } \widehat{\mathscr{B}} \in \mathscr{L}^{\mathrm{q}}_{\mathtt{m},0}, \quad \text{such that} \quad \mathscr{D} \subset \widehat{\mathscr{B}} \quad \Longleftrightarrow \quad \operatorname{rank}\bigl(\Phi(\mathscr{D})\bigr) \leq \mathtt{m}. \qquad (*)$$

The exact model

$$\widehat{\mathscr{B}} = \operatorname{image}\bigl(\Phi(\mathscr{D})\bigr) \qquad (**)$$

of minimal dimension

$$\mathtt{c}(\mathscr{B}) = \operatorname{rank}\bigl(\Phi(\mathscr{D})\bigr)$$

always exists and is unique.

The equivalence $(*)$ between data modeling and the concept of rank is the basis for application of linear algebra and matrix computations to linear data modeling. Indeed, $(**)$ provides an *algorithm* for exact linear static data modeling. As shown next, exact data modeling has also direct relevance to approximate data modeling.

Approximate Data Modeling

When an exact model does not exist in the considered model class, an approximate model that is in some sense "close" to the data is aimed at instead. Closeness is measured by a suitably defined criterion. This leads to the following approximate data modeling problem.

Problem 2.16 (Approximate data modeling) Given data $\mathscr{D} \subset \mathscr{U}$, a model class $\mathscr{M}_{\mathtt{c}_{\max}} \in 2^{\mathscr{U}}$, and a measure $f(\mathscr{D}, \mathscr{B})$ for the lack of fit of the data $\mathscr{D}$ by a model $\mathscr{B}$, find a model $\widehat{\mathscr{B}}$ in the model class $\mathscr{M}_{\mathtt{c}_{\max}}$ that minimizes the lack of fit, i.e.,

$$\text{minimize} \quad \text{over } \mathscr{B} \in \mathscr{M}_{\mathtt{c}_{\max}} \quad f(\mathscr{D}, \mathscr{B}). \qquad \text{(AM)}$$

Since an observation w is a point in and the model $\mathscr{B}$ is a subset of the data space $\mathscr{U}$, it is natural to measure the lack of fit between w and $\mathscr{B}$ by the *geometric distance*

$$\operatorname{dist}(w, \mathscr{B}) := \min_{\widehat{w} \in \mathscr{B}} \bigl\| w - \widehat{w} \bigr\|_2. \qquad (\operatorname{dist}(w, \mathscr{B}))$$

The auxiliary variable $\widehat{w}$ is the best approximation of w in $\mathscr{B}$. Geometrically, it is the orthogonal projection of w on $\mathscr{B}$.

For the set of observations $\mathscr{D}$, we define the distance from $\mathscr{D}$ to $\mathscr{B}$ as

$$\operatorname{dist}(\mathscr{D}, \mathscr{B}) := \min_{\widehat{w}_1, \ldots, \widehat{w}_N \in \mathscr{B}} \sqrt{\sum_{j=1}^{N} \bigl\| w_{\mathrm{d},j} - \widehat{w}_j \bigr\|_2^2}. \qquad (\operatorname{dist})$$

The set of points

$$\widehat{\mathscr{D}} = \{\widehat{w}_1, \ldots, \widehat{w}_N\}$$

in the definition of (dist) is an approximation of the data $\mathscr{D}$ in the model $\mathscr{B}$. Note that problem (dist) is separable, i.e., it decouples into N independent problems (dist($w, \mathscr{B}$)).

Algorithms for computing the geometric distance are discussed in Sect. 3.2, in the case of linear models, and in Chap. 6, in the case of polynomial models.

Note 2.17 (Invariance of (dist) to rigid transformation) The geometric distance $\operatorname{dist}(\mathscr{D}, \mathscr{B})$ is invariant to a rigid transformation, i.e., translation, rotation, and reflection of the data points and the model.

An alternative distance measure, called *algebraic distance*, is based on a kernel representation $\mathscr{B} = \ker(R)$ of the model $\mathscr{B}$. Since R is a mapping from $\mathscr{U}$ to $\mathbb{R}^g$, such that

$$w \in \mathscr{B} \quad \Longleftrightarrow \quad R(w) = 0,$$

we have

$$\left\| R(w) \right\|_{\mathrm{F}} > 0 \quad \Longleftrightarrow \quad w \notin \mathscr{B}.$$

The algebraic "distance" measures the lack of fit between w and $\mathscr{B}$ by the "size" $\|R(w)\|_{\mathrm{F}}$ of the residual $R(w)$. For a data set $\mathscr{D}$, we define

$$\operatorname{dist}'(\mathscr{D}, \mathscr{B}) := \sqrt{\sum_{j=1}^{N} \left\| R(w_{\mathrm{d},j}) \right\|_{\mathrm{F}}^2}. \tag{dist$'$}$$

The algebraic distance depends on the choice of the parameter R in a kernel representation of the model, while the geometric distance is representation invariant. In addition, the algebraic distance is not invariant to a rigid transformation. However, a modification of the algebraic distance that is invariant to a rigid transformation is presented in Sect. 6.3.

Example 2.18 (Geometric distance for linear and quadratic models) The two plots in Fig. 2.3 illustrate the geometric distance (dist) from a set of eight data points

$$\mathscr{D} = \{d_i = (x_i, y_i) \mid i = 1, \ldots, 8\}$$

in the plane to, respectively, linear $\mathscr{B}_1$ and quadratic $\mathscr{B}_2$ models. As its name suggests, $\operatorname{dist}(\mathscr{D}, \mathscr{B})$ has geometric interpretation—in order to compute the geometric distance, we project the data points on the models. This is a simple task (linear least squares problem) for linear models but a nontrivial task (nonconvex optimization problem) for nonlinear models. In contrast, the algebraic "distance" (not visualized in the figure) has no simple geometrical interpretation but is easy to compute for linear and nonlinear models alike.

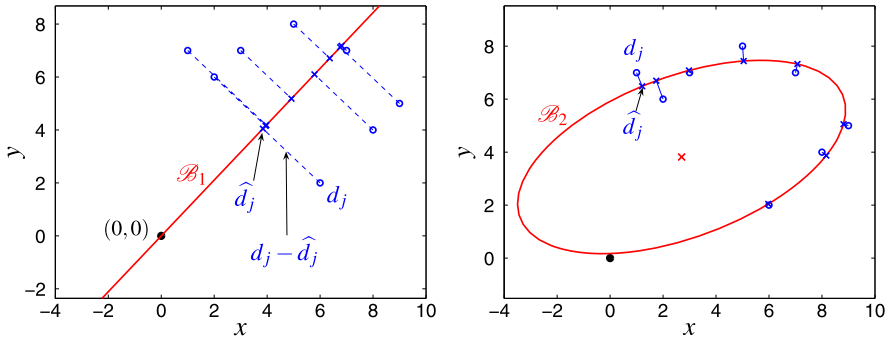


Fig. 2.3 Geometric distance from eight data points to a linear (*left*) and quadratic (*right*) models

Note 2.19 (Approximate modeling in the case of exact data) If an exact model $\mathscr{B}$ exists in the model class $\mathscr{M}_{\mathsf{c}_{\max}}$, then $\mathscr{B}$ is a global optimum point of the approximate modeling problem (AM) (irrespective of the approximation criterion f being used). Indeed,

$$\mathscr{D} \subset \mathscr{B} \quad \Longleftrightarrow \quad \operatorname{dist}(\mathscr{D}, \mathscr{B}) = \operatorname{dist}'(\mathscr{D}, \mathscr{B}) = 0.$$

An optimal approximate model, i.e., a solution of (AM), however, need not be unique. In contrast, the most powerful unfalsified model is unique. This is due to the fact that (AM) imposes an upper bound but does not minimize the model complexity, while (EM) minimizes the model complexity. As a result, when

$$\mathsf{c}\big(\mathscr{B}_{\mathrm{mpum}}(\mathscr{D})\big) < \mathsf{c}_{\max},$$

(AM) has a nonunique solution. In the next section, we present a more general approximate data modeling problem formulation that minimizes simultaneously the complexity as well as the fitting error.

The terminology "geometric" and "algebraic" distance comes from the computer vision application of the methods for fitting curves and surfaces to data. In the system identification community, the geometric fitting method is related to the *misfit* approach and the algebraic fitting criterion is related to the *latency* approach. Misfit and latency computation are data smoothing operations. For linear time-invariant systems, the misfit and latency can be computed efficiently by Riccati type recursions. In the statistics literature, the geometric fitting is related to errors-in-variable estimation and the algebraic fitting is related to classical regression estimation, see Table 2.2.

Example 2.20 (Algebraic fit and errors-in-variables modeling) From a statistical point of view, the approximate data modeling problem (AM) with the geometric fitting criterion (dist) yields a maximum likelihood estimator for the true model $\mathscr{B}_0$ in the errors-in-variables setup

$$w_{\mathrm{d},j} = w_{0,j} + \widetilde{w}_j, \tag{EIV}$$

Table 2.2 Correspondence among terms for data fitting criteria in different fields

Computer vision	System identification	Statistics	Mathematics
geometric fitting	misfit	errors-in-variables	implicit function
algebraic fitting	latency	regression	function

where

$$\mathscr{D}_0 := \{w_{0,1}, \ldots, w_{0,N}\} \subset \mathscr{B}_0$$

is the true data and

$$\widetilde{\mathscr{D}} := \{\widetilde{w}_1, \ldots, \widetilde{w}_N\}$$

is the measurement noise, which is assumed to be a set of independent, zero mean, Gaussian random vectors, with covariance matrix $\sigma^2 I$.

Example 2.21 (Algebraic fit by a linear model and regression) A linear model class, defined by the input/output representation $\mathscr{B}_{\mathrm{i/o}}(\Theta)$ and algebraic fitting criterion (dist′), where

$$w := \mathrm{col}(u, y) \quad \text{and} \quad R(w) := \Theta^\top u - y$$

lead to the ordinary linear least squares problem

$$\text{minimize} \quad \text{over } \Theta \in \mathbb{R}^{\mathtt{m}\times\mathtt{p}} \quad \left\| \Theta^\top \Phi(u_{\mathrm{d}}) - \Phi(y_{\mathrm{d}}) \right\|_{\mathrm{F}}. \tag{LS}$$

The statistical setting for the least squares approximation problem (LS) is the classical regression model

$$R(w_{\mathrm{d},j}) = e_j, \tag{REG}$$

where $e_1, \ldots, e_N$ are zero mean independent and identically distributed random variables. Gauss–Markov's theorem states that the least squares approximate solution is the best linear unbiased estimator for the regression model (REG).

Complexity–Accuracy Trade-off

Data modeling is a mapping from a given data set $\mathscr{D}$, to a model $\mathscr{B}$ in a given model class $\mathscr{M}$:

$$\text{data set } \mathscr{D} \subset \mathscr{U} \xrightarrow{\text{data modeling problem}} \text{model } \mathscr{B} \in \mathscr{M} \in 2^{\mathscr{U}}.$$

A data modeling problem is defined by specifying the model class $\mathscr{M}$ and one or more modeling criteria. Basic criteria in any data modeling problem are:

- "simple" model, measured by the model complexity $\mathrm{c}(\mathscr{B})$, and
- "good" fit of the data by the model, measured by (vector) cost function $F(\mathscr{D}, \mathscr{B})$.

Small complexity $\mathrm{c}(\mathscr{B})$ and small fitting error $F(\mathscr{D}, \mathscr{B})$, however, are contradicting objectives, so that a core issue throughout data modeling is the complexity–accuracy trade-off. A generic data modeling problem is:

Given a data set $\mathscr{D} \in \mathscr{U}$ and a measure F for the fitting error, solve the multi-objective optimization problem:

$$\text{minimize} \quad \text{over } \mathscr{B} \in \mathscr{M} \quad \begin{bmatrix} \mathrm{c}(\mathscr{B}) \\ F(\mathscr{D}, \mathscr{B}) \end{bmatrix}. \tag{DM}$$

Next we consider the special cases of linear static models and linear time-invariant dynamic models with $F(\mathscr{D}, \mathscr{B})$ being $\mathrm{dist}(\mathscr{D}, \mathscr{B})$ or $\mathrm{dist}'(\mathscr{D}, \mathscr{B})$. The model class assumption implies that $\dim(\mathscr{B})$ is a complexity measure in both static and dynamic cases.

Two Possible Scalarizations: Low Rank Approximation and Rank Minimization

The data set $\mathscr{D}$, can be parametrized by a real vector $p \in \mathbb{R}^{n_p}$. (Think of the vector p as a representation of the data in the computer memory.) For a linear model $\mathscr{B}$ and exact data $\mathscr{D}$, there is a relation between the model complexity and the rank of a data matrix $\mathscr{S}(p)$:

$$\mathrm{c}\big(\mathscr{B}_{\mathrm{mpum}}(\mathscr{D})\big) = \mathrm{rank}\big(\mathscr{S}(p)\big). \tag{$*$}$$

The mapping

$$\mathscr{S} : \mathbb{R}^{n_p} \to \mathbb{R}^{m \times n}$$

from the data parameter vector p to the data matrix $\mathscr{S}(p)$ depends on the application. For example, $\mathscr{S}(p) = \Phi(\mathscr{D})$ is unstructured in the case of linear static modeling (see Example 2.15) and $\mathscr{S}(p) = \mathscr{H}(w_{\mathrm{d}})$ is Hankel structured in the case of autonomous linear time-invariant dynamic model identification,

Let p be the parameter vector for the data $\mathscr{D}$ and $\widehat{p}$ be the parameter vector for the data approximation $\widehat{\mathscr{D}}$. The geometric distance $\mathrm{dist}(\mathscr{D}, \mathscr{B})$ can be expressed in terms of the parameter vectors p and $\widehat{p}$ as

$$\text{minimize} \quad \text{over } \widehat{p} \quad \|p - \widehat{p}\|_2 \quad \text{subject to} \quad \widehat{\mathscr{D}} \subset \mathscr{B}.$$

Moreover, the norm in the parameter space $\mathbb{R}^{n_p}$ can be chosen as weighted 1-, 2-, and ∞-(semi)norms:

$$\begin{aligned}
\|\widetilde{p}\|_{w,1} &:= \left\|w \odot \widetilde{p}\right\|_1 := \sum_{i=1}^{n_p} \left|w_i \widetilde{p}_i\right|, \\
\|\widetilde{p}\|_{w,2} &:= \left\|w \odot \widetilde{p}\right\|_2 := \sqrt{\sum_{i=1}^{n_p} \left(w_i \widetilde{p}\right)^2}, \qquad (\|\cdot\|_w) \\
\|\widetilde{p}\|_{w,\infty} &:= \left\|w \odot \widetilde{p}\right\|_\infty := \max_{i=1,\ldots,n_p} \left|w_i \widetilde{p}_i\right|,
\end{aligned}$$

where w is a vector with nonnegative elements, specifying the weights, and $\odot$ is the element-wise (Hadamard) product.

Using the data parametrization $(*)$ and one of the distance measures $(\|\cdot\|_w)$, the data modeling problem (DM) becomes the biobjective matrix approximation problem:

$$\text{minimize} \quad \text{over } \widehat{p} \quad \begin{bmatrix} \operatorname{rank}(\mathscr{S}(\widehat{p})) \\ \|p - \widehat{p}\| \end{bmatrix}. \tag{DM'}$$

Two possible ways to scalarize the biobjective problem (DM') are:

1. Misfit minimization subject to a bound r on the model complexity

$$\text{minimize} \quad \text{over } \widehat{p} \quad \left\|p - \widehat{p}\right\| \quad \text{subject to} \quad \operatorname{rank}\left(\mathscr{S}(\widehat{p})\right) \leq r. \tag{SLRA}$$

2. Model complexity minimization subject to a bound ε on the fitting error

$$\text{minimize} \quad \text{over } \widehat{p} \quad \operatorname{rank}\left(\mathscr{S}(\widehat{p})\right) \quad \text{subject to} \quad \left\|p - \widehat{p}\right\| \leq \varepsilon. \tag{RM}$$

Problem (SLRA) is a structured low rank approximation problem and (RM) is a rank minimization problem.

By varying the parameters r and ε from zero to infinity, both problems sweep the trade-off curve (set of Pareto optimal solutions) of (DM'). Note, however, that r ranges over the natural numbers and only small values are of practical interest. In addition, in applications often a "suitable" value for r can be chosen a priori or is even a part of the problem specification. In contrast, ε is a positive real number and is data dependent, so that a "suitable" value is not readily available. These considerations, suggest that the structured low rank approximation problem is a more convenient scalarization of (DM') for solving practical data modeling problems.

Convex relaxation algorithms for solving (DM') are presented in Sect. 3.3.

2.4 Unstructured Low Rank Approximation

Linear static data modeling leads to unstructured low rank approximation. Vice verse, unstructured low rank approximation problems can be given the interpretation (or motivation) of linear static data modeling problems. As argued in Sect. 1.1, these are equivalent problems. The data modeling view of the problem makes link

to applications. The low rank approximation view of the problem makes link to computational algorithms for solving the problem.

Problem 2.22 (Unstructured low rank approximation) Given a matrix $D \in \mathbb{R}^{\mathtt{q} \times N}$, with $\mathtt{q} \leq N$, a matrix norm $\|\cdot\|$, and an integer $\mathtt{m}$, $0 < \mathtt{m} < \mathtt{q}$, find a matrix

$$\widehat{D}^* := \arg\min_{\widehat{D}} \left\| D - \widehat{D} \right\| \quad \text{subject to} \quad \operatorname{rank}(\widehat{D}) \leq \mathtt{m}. \tag{LRA}$$

The matrix $\widehat{D}^*$ is an optimal rank-$\mathtt{m}$ (or less) approximation of D with respect to the given norm $\|\cdot\|$.

The special case of (LRA) with $\|\cdot\|$ being the weighted 2-norm

$$\|\Delta D\|_W := \sqrt{\operatorname{vec}^\top(\Delta D) W \operatorname{vec}(\Delta D)}, \quad \text{for all } \Delta D \tag{$\|\cdot\|_W$}$$

where $W \in \mathbb{R}^{\mathtt{q}N \times \mathtt{q}N}$ is a positive definite matrix, is called *weighted low rank approximation problem*. In turn, special cases of the weighted low rank approximation problem are obtained when the weight matrix W has diagonal, block diagonal, or some other structure.

- *Element-wise weighting:*

$$W = \operatorname{diag}(w_1, \ldots, w_{\mathtt{q}N}), \quad \text{where } w_i > 0, \text{ for } i = 1, \ldots, \mathtt{q}N.$$

- *Column-wise weighting:*

$$W = \operatorname{diag}(W_1, \ldots, W_N), \quad \text{where } W_j \in \mathbb{R}^{\mathtt{q} \times \mathtt{q}}, \ W_j > 0, \text{ for } j = 1, \ldots, N.$$

- *Column-wise weighting with equal weight matrix for all columns:*

$$W = \operatorname{diag}(W_1, \ldots, W_1), \quad \text{where } W_1 \in \mathbb{R}^{\mathtt{q} \times \mathtt{q}}, \ W_1 > 0.$$

- *Row-wise weighting:*

$$\overline{W} = \operatorname{diag}(W_1, \ldots, W_{\mathtt{q}}), \quad \text{where } W_i \in \mathbb{R}^{N \times N}, \ W_i > 0, \text{ for } i = 1, \ldots, \mathtt{q},$$

and $\overline{W}$ is a matrix, such that

$$\|\Delta D\|_W = \left\| \Delta D^\top \right\|_{\overline{W}}, \quad \text{for all } \Delta D.$$

- *Row-wise weighting with equal weight matrix for all rows:*

$$\overline{W} = \operatorname{diag}(\underbrace{W_{\mathtt{r}}, \ldots, W_{\mathtt{r}}}_{\mathtt{q}}), \quad \text{where } W_{\mathtt{r}} \in \mathbb{R}^{N \times N}, \ W_{\mathtt{r}} > 0.$$

Figure 2.4 shows the hierarchy of weighted low rank approximation problems, according to the structure of the weight matrix W. Exploiting the structure of the weight matrix allows more efficient solution of the corresponding weighted low rank approximation problems compared to the general problem with unstructured W.

As shown in the next section left/right weighting with equal matrix for all rows/columns corresponds to the approximation criteria $\|\sqrt{W_l}\Delta D\|_F$ and $\|\Delta D\sqrt{W_r}\|_F$. The approximation problem with criterion $\|\sqrt{W_l}\Delta D\sqrt{W_r}\|_F$ is called *two-sided weighted* and is also known as the generalized low rank approximation problem. This latter problem allows analytic solution in terms of the singular value decomposition of the data matrix.

Special Cases with Known Analytic Solutions

An extreme special case of the weighted low rank approximation problem is the "unweighted" case, i.e., weight matrix a multiple of the identity $W = v^{-1}I$, for some $v > 0$. Then, $\|\cdot\|_W$ is proportional to the Frobenius norm $\|\cdot\|_F$ and the low rank approximation problem has an analytic solution in terms of the singular value decomposition of D. The results is known as the Eckart–Young–Mirsky theorem or the matrix approximation lemma. In view of its importance, we refer to this case as the *basic low rank approximation problem*.

Theorem 2.23 (Eckart–Young–Mirsky) *Let*

$$D = U\Sigma V^\top$$

be the singular value decomposition of D and partition U, $\Sigma =: \operatorname{diag}(\sigma_1, \ldots, \sigma_q)$, and V as follows:

$$\overset{\begin{matrix}m & q-m\end{matrix}}{U =: \begin{bmatrix} U_1 & U_2 \end{bmatrix}}\begin{matrix} \\ q \end{matrix}, \quad \overset{\begin{matrix}m & q-m\end{matrix}}{\Sigma =: \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix}}\begin{matrix} m \\ q-m \end{matrix} \quad \textit{and} \quad \overset{\begin{matrix}m & q-m\end{matrix}}{V =: \begin{bmatrix} V_1 & V_2 \end{bmatrix}}\begin{matrix} \\ N \end{matrix}.$$

Then the rank-m matrix, obtained from the truncated singular value decomposition

$$\widehat{D}^* = U_1\Sigma_1 V_1^\top,$$

is such that

$$\|D - \widehat{D}^*\|_F = \min_{\operatorname{rank}(\widehat{D})\le m} \|D - \widehat{D}\|_F = \sqrt{\sigma_{m+1}^2 + \cdots + \sigma_q^2}.$$

The minimizer $\widehat{D}^*$ is unique if and only if $\sigma_{m+1} \neq \sigma_m$.

The proof is given in Appendix B.

Note 2.24 (Unitarily invariant norms) Theorem 2.23 holds for any norm $\|\cdot\|$ that is invariant under orthogonal transformations, i.e., satisfying the relation

$$\|U\Delta DV\| = \|\Delta D\|, \quad \text{for any } \Delta D \text{ and for any orthogonal matrices } U \text{ and } V.$$

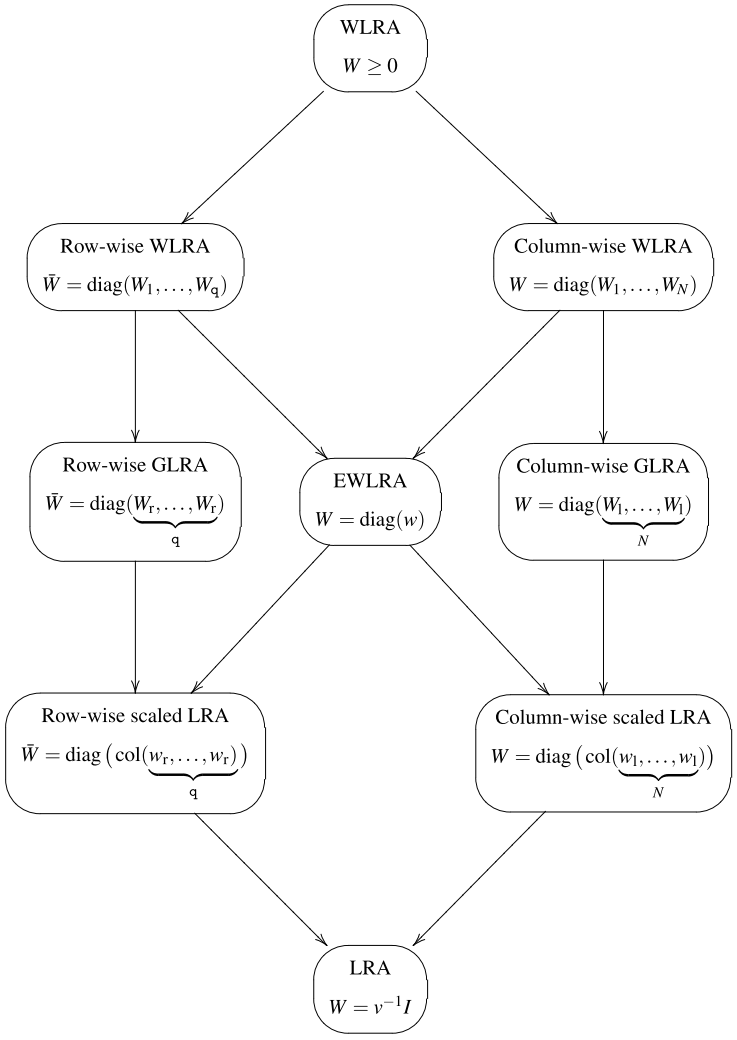


Fig. 2.4 Hierarchy of weighted low rank approximation problems according to the structure of the weight matrix W. On the *left side* are weighted low rank approximation problems with row-wise weighting and on the *right side* are weighted low rank approximation problems with column-wise weighting. The generality of the problem reduces from *top* to *bottom*

Note 2.25 (Approximation in the spectral norm) For a matrix ΔD, let $\|\Delta D\|_2$ be the spectral (2-norm induced) matrix norm

$$\|\Delta D\|_2 = \sigma_{\max}(\Delta D).$$

Then

$$\min_{\operatorname{rank}(\widehat{D})=\mathtt{m}} \left\| D - \widehat{D} \right\|_2 = \sigma_{\mathtt{m}+1},$$

i.e., the optimal rank-$\mathtt{m}$ spectral norm approximation error is equal to the first neglected singular value. The truncated singular value decomposition yields an optimal approximation with respect to the spectral norm, however, in this case a minimizer is not unique even when the singular values $\sigma_{\mathtt{m}}$ and $\sigma_{\mathtt{m}+1}$ are different.

As defined, the low rank approximation problem aims at a matrix $\widehat{D}$ that is a solution to the optimization problem (LRA). In data modeling problems, however, of primary interest is the optimal model, i.e., the most powerful unfalsified model for $\widehat{D}^*$. Theorem 2.23 gives the optimal approximating matrix $\widehat{D}^*$ in terms of the singular value decomposition of the data matrix D. Minimal parameters of kernel and image representations of the corresponding optimal model are directly available from the factors of the singular value decomposition of D.

Corollary 2.26 *An optimal in the Frobenius norm approximate model for the data D in the model class $\mathscr{L}_{\mathtt{m},0}$, i.e., $\widehat{\mathscr{B}}^* := \mathscr{B}_{\mathrm{mpum}}(\widehat{D}^*)$ is unique if and only if the singular values $\sigma_{\mathtt{m}}$ and $\sigma_{\mathtt{m}+1}$ of D are different, in which case*

$$\widehat{\mathscr{B}}^* = \ker\left(U_2^\top\right) = \operatorname{image}(U_1).$$

The proof is left as an exercise.

64 ⟨*Low rank approximation* 64⟩≡

```
function [R, P, dh] = lra(d, r)
[u, s, v] = svd(d, 0); R = u(:, (r + 1):end)';
P = u(:, 1:r);
if nargout > 2, dh = u(:, 1:r) * s(1:r, 1:r) * v(:, 1:r)';
end
```

Defines:
lra, used in chunks 85d, 88c, 101, 191b, 192e, and 229a.

Corollary 2.27 (Nested approximations) *The optimal in the Frobenius norm approximate models $\widehat{\mathscr{B}}^*_{\mathtt{m}}$ for the data D in the model classes $\mathscr{L}_{\mathtt{m},0}$, where $\mathtt{m} = 1, \ldots, \mathtt{q}$ are nested, i.e.,*

$$\widehat{\mathscr{B}}_{\mathtt{q}} \subseteq \widehat{\mathscr{B}}_{\mathtt{q}-1} \subseteq \cdots \subset \widehat{\mathscr{B}}_1.$$

The proof is left as an exercise.

Note 2.28 (Efficient computation using QR factorization when $N \gg \mathtt{q}$) An optimal model $\widehat{\mathscr{B}}^*$ for D depends only on the left singular vectors of D. Since post multiplication of D by an orthogonal matrix Q does not change the left singular vectors $\widehat{\mathscr{B}}^*$ is an optimal model for the data matrix DQ.

For $N \gg \mathfrak{q}$, computing the QR factorization

$$D^\top = \begin{bmatrix} R_1 \\ 0 \end{bmatrix} Q^\top, \quad \text{where } R_1 \text{ is upper triangular,} \tag{QR}$$

and the singular value decomposition of R_1 is a more efficient alternative for finding $\widehat{\mathscr{B}}$ than computing the singular value decomposition of D.

An analytic solution in terms of the singular value decomposition, similar to the one in Theorem 2.23 is not known for the general weighted low rank approximation problem. Presently the largest class of weighted low rank approximation problems with analytic solution are those with a weight matrix of the form

$$W = W_{\mathrm{r}} \otimes W_{\mathrm{l}}, \quad \text{where } W_{\mathrm{l}} \in \mathbb{R}^{\mathfrak{q}\times\mathfrak{q}} \text{ and } W_{\mathrm{r}} \in \mathbb{R}^{N\times N} \tag{$W_{\mathrm{r}} \otimes W_{\mathrm{l}}$}$$

are positive definite matrices and $\otimes$ is the Kronecker product.

Using the identities

$$\operatorname{vec}(AXB) = \left(B^\top \otimes A\right) \operatorname{vec}(X)$$

and

$$(A_1 \otimes B_1)(A_2 \otimes B_2) = (A_1 A_2) \otimes (B_1 B_2),$$

we have

$$\begin{aligned} \left\| D - \widehat{D} \right\|_{W_{\mathrm{r}} \otimes W_{\mathrm{l}}} &= \sqrt{\operatorname{vec}^\top(\Delta D)(W_{\mathrm{r}} \otimes W_{\mathrm{l}}) \operatorname{vec}(\Delta D)} \\ &= \left\| \left(\sqrt{W_{\mathrm{r}}} \otimes \sqrt{W_{\mathrm{l}}}\right) \operatorname{vec}(\Delta D) \right\|_2 \\ &= \left\| \operatorname{vec}\left(\sqrt{W_{\mathrm{l}}} \Delta D \sqrt{W_{\mathrm{r}}}\right) \right\|_2 \\ &= \left\| \sqrt{W_{\mathrm{l}}} \Delta D \sqrt{W_{\mathrm{r}}} \right\|_{\mathrm{F}}. \end{aligned}$$

Therefore, the low rank approximation problem (LRA) with norm $(\|\cdot\|_W)$ and weight matrix $(W_{\mathrm{r}} \otimes W_{\mathrm{l}})$ is equivalent to the two-sided weighted (or generalized) low rank approximation problem

$$\begin{aligned} \text{minimize} \quad & \text{over } \widehat{D} \quad \left\| \sqrt{W_{\mathrm{l}}} \left(D - \widehat{D}\right) \sqrt{W_{\mathrm{r}}} \right\|_{\mathrm{F}} \\ \text{subject to} \quad & \operatorname{rank}\left(\widehat{D}\right) \leq \mathfrak{m}, \end{aligned} \tag{WLRA2}$$

which has an analytic solution.

Theorem 2.29 (Two-sided weighted low rank approximation) *Define the modified data matrix*

$$D_{\mathrm{m}} := \sqrt{W_{\mathrm{l}}} D \sqrt{W_{\mathrm{r}}},$$

and let $\widehat{D}^*_{\mathrm{m}}$ *be the optimal* (*unweighted*) *low rank approximation of* D_{m}. *Then*

$$\widehat{D}^* := \left(\sqrt{W_{\mathrm{l}}}\right)^{-1} \widehat{D}^*_{\mathrm{m}} \left(\sqrt{W_{\mathrm{r}}}\right)^{-1},$$

is a solution of the following two-sided weighted low rank approximation problem (WLRA2). *A solution always exists. It is unique if and only if* $\widehat{D}^*_{\mathrm{m}}$ *is unique.*

The proof is left as an exercise (Problem P.14).

Exercise 2.30 Using the result in Theorem 2.29, write a function that solves the two-sided weighted low rank approximation problem (WLRA2).

Data Modeling via (LRA)

The following problem is the approximate modeling problem (AM) for the model class of linear static models, i.e., $\mathscr{M}_{\mathtt{c}_{\max}} = \mathscr{L}_{\mathtt{m},0}$, with the orthogonal distance approximation criterion, i.e., $f(\mathscr{D}, \mathscr{B}) = \operatorname{dist}(\mathscr{D}, \mathscr{B})$. The norm $\|\cdot\|$ in the definition of dist, however, in the present context is a general vector norm, rather than the 2-norm.

Problem 2.31 (Static data modeling) Given N, $\mathtt{q}$-variable observations

$$\{d_1, \ldots, d_N\} \subset \mathbb{R}^{\mathtt{q}},$$

a matrix norm $\|\cdot\|$, and model complexity $\mathtt{m}$, $0 < \mathtt{m} < \mathtt{q}$,

$$\begin{array}{lll} \text{minimize} & \text{over } \widehat{\mathscr{B}} \text{ and } \widehat{D} & \left\|D - \widehat{D}\right\| \\ \text{subject to} & \operatorname{image}(\widehat{D}) \subseteq \widehat{\mathscr{B}} \quad \text{and} & \dim(\widehat{\mathscr{B}}) \leq \mathtt{m}, \end{array} \qquad (\text{AM } \mathscr{L}_{\mathtt{m},0})$$

where $D \in \mathbb{R}^{\mathtt{q} \times N}$ is the data matrix $D := [d_1 \ \cdots \ d_N]$.

A solution $\widehat{\mathscr{B}}^*$ to (AM $\mathscr{L}_{\mathtt{m},0}$) is an optimal approximate model for the data D with complexity bounded by $\mathtt{m}$. Of course, $\widehat{\mathscr{B}}^*$ depends on the approximation criterion, specified by the given norm $\|\cdot\|$. A justification for the choice of the norm $\|\cdot\|$ is provided in the errors-in-variables setting (see Example 2.20), i.e., the data matrix D is assumed to be a noisy measurement of a true matrix D_0

$$\begin{array}{l} D = D_0 + \widetilde{D}, \quad \operatorname{image}(D_0) = \mathscr{B}_0, \quad \dim(\mathscr{B}_0) \leq \mathtt{m}, \quad \text{and} \\ \operatorname{vec}(\widetilde{D}) \sim \mathrm{N}(0, \sigma^2 W^{-1}), \quad \text{where } W \succ 0 \end{array} \qquad (\text{EIV}_0)$$

and $\widetilde{D}$ is the measurement error that is assumed to be a random matrix with zero mean and normal distribution. The true matrix D_0 is "generated" by a model $\mathscr{B}_0$, with a known complexity bound $\mathtt{m}$. The model $\mathscr{B}_0$ is the object to be estimated in the errors-in-variables setting.

Proposition 2.32 (Maximum likelihood property of optimal static model $\widehat{\mathscr{B}}^*$) *Assume that the data are generated in the errors-in-variables setting* (EIV_0), *where the matrix* $W \succ 0$ *is known and the scalar* σ^2 *is unknown. Then a solution* $\widehat{\mathscr{B}}^*$ *to Problem 2.31 with weighted 2-norm* ($\|\cdot\|_W$) *is a maximum likelihood estimator for the true model* $\mathscr{B}_0$.

The proof is given in Appendix B.

The main assumption of Proposition 2.32 is

$$\operatorname{cov}\big(\operatorname{vec}(\widetilde{D})\big) = \sigma^2 W^{-1}, \quad \text{with } W \text{ given.}$$

Note, however, that σ^2 is not given, so that the probability density function of $\widetilde{D}$ is not completely specified. Proposition 2.32 shows that the problem of computing the maximum likelihood estimator in the errors-in-variables setting is equivalent to Problem 2.22 with the weighted norm $\|\cdot\|_W$. Maximum likelihood estimation for density functions other than normal leads to low rank approximation with norms other than the weighted 2-norm.

2.5 Structured Low Rank Approximation

Structured low rank approximation is a low rank approximation, in which the approximating matrix $\widehat{D}$ is constrained to have some a priori specified structure; typically, the same structure as the one of the data matrix D. Common structures encountered in applications are Hankel, Toeplitz, Sylvester, and circulant as well as their block versions. In order to state the problem in its full generality, we first define a structured matrix. Consider a mapping $\mathscr{S}$ from a parameter space $\mathbb{R}^{n_p}$ to a set of matrices $\mathbb{R}^{m\times n}$. A matrix $\widehat{D} \in \mathbb{R}^{m\times n}$ is called $\mathscr{S}$*-structured* if it is in the image of $\mathscr{S}$, i.e., if there exists a parameter $\widehat{p} \in \mathbb{R}^{n_p}$, such that $\widehat{D} = \mathscr{S}(\widehat{p})$.

Problem SLRA (Structured low rank approximation) Given a structure specification

$$\mathscr{S} : \mathbb{R}^{n_p} \to \mathbb{R}^{m\times n}, \quad \text{with } m \leq n,$$

a parameter vector $p \in \mathbb{R}^{n_p}$, a vector norm $\|\cdot\|$, and an integer r, $0 < r < \min(m, n)$,

$$\text{minimize} \quad \text{over } \widehat{p} \quad \big\|p - \widehat{p}\big\| \quad \text{subject to} \quad \operatorname{rank}\big(\mathscr{S}(\widehat{p})\big) \leq r. \qquad \text{(SLRA)}$$

The matrix $\widehat{D}^* := \mathscr{S}(\widehat{p}^*)$ is an optimal rank-r (or less) approximation of the matrix $D := \mathscr{S}(p)$, within the class of matrices with the same structure as D. Problem SLRA is a generalization of Problem 2.22. Indeed, choosing $\mathscr{S}$ to be vec^{-1} (an operation reconstructing a matrix M from the vector $\operatorname{vec}(M)$) and norm $\|\Delta p\|$ to be such that

$$\|\Delta p\| = \big\|\mathscr{S}(\Delta p)\big\|, \quad \text{for all } \Delta p,$$

Problem SLRA is equivalent to Problem 2.22.

Special Cases with Known Analytic Solutions

We showed that some weighted unstructured low rank approximation problems have global analytic solution in terms of the singular value decomposition. Similar result exists for circulant structured low rank approximation. If the approximation criterion is a unitarily invariant matrix norm, the unstructured low rank approximation (obtained for example from the truncated singular value decomposition) is unique. In the case of a circulant structure, it turns out that this unique minimizer also has circulant structure, so the structure constraint is satisfied without explicitly enforcing it in the approximation problem.

An efficient computational way of obtaining the circulant structured low rank approximation is the fast Fourier transform. Consider the scalar case and let

$$P_k := \sum_{j=1}^{n_p} p_j e^{-\mathbf{i}\frac{2\pi}{n_p}kj}$$

be the discrete Fourier transform of p. Denote with $\mathscr{K}$ the subset of $\{1, \ldots, n_p\}$ consisting of the indices of the $\mathtt{m}$ largest elements of $\{|P_1|, \ldots, |P_{n_p}|\}$. Assuming that $\mathscr{K}$ is uniquely defined by the above condition, i.e., assuming that

$$k \in \mathscr{K} \quad \text{and} \quad k' \notin \mathscr{K} \quad \Longrightarrow \quad |P_k| > |P_{k'}|,$$

the solution $\widehat{p}^*$ of the structured low rank approximation problem with $\mathscr{S}$ a circulant matrix is unique and is given by

$$\widehat{p}^* = \frac{1}{n_p} \sum_{k \in \mathscr{K}} P_k e^{\mathbf{i}\frac{2\pi}{n_p}kj}.$$

Data Modeling via (SLRA)

The reason to consider the more general structured low rank approximation is that $D = \mathscr{S}(p)$ being low rank and Hankel structured is equivalent to p being generated by a linear time-invariant dynamic model. To show this, consider first the special case of a scalar Hankel structure

$$\mathscr{H}_{1+1}(p) := \begin{bmatrix} p_1 & p_2 & \cdots & p_{n_p-1} \\ p_2 & p_3 & \cdots & p_{n_p-1+1} \\ \vdots & \vdots & \ddots & \vdots \\ p_{1+1} & p_{1+2} & \cdots & p_{n_p} \end{bmatrix}.$$

The approximation matrix

$$\widehat{D} = \mathscr{H}_{1+1}(\widehat{p})$$

being rank deficient implies that there is a nonzero vector $R = [R_0 \; R_1 \; \cdots \; R_{\mathtt{l}}]$, such that

$$R\mathscr{H}_{\mathtt{l}+1}(\widehat{p}) = 0.$$

Due to the Hankel structure, this system of equations can be written as

$$R_0\widehat{p}_t + R_1\widehat{p}_{t+1} + \cdots + R_{\mathtt{l}}\widehat{p}_{t+\mathtt{l}} = 0, \quad \text{for } t = 1, \ldots, n_p - \mathtt{l},$$

i.e., a homogeneous constant coefficients difference equation. Therefore, $\widehat{p}$ is a trajectory of an autonomous linear time-invariant system, defined by (KER). Recall that for an autonomous system $\mathscr{B}$,

$$\dim(\mathscr{B}|_{[1,T]}) = \mathtt{n}(\mathscr{B}), \quad \text{for } T \geq \mathtt{n}(\mathscr{B}).$$

The scalar Hankel low rank approximation problem is then equivalent to the following dynamic modeling problem. Given T samples of a scalar signal $w_{\mathrm{d}} \in \mathbb{R}^T$, a signal norm $\|\cdot\|$, and a model complexity $\mathtt{n}$,

$$\begin{array}{ll} \text{minimize} & \text{over } \widehat{\mathscr{B}} \text{ and } \widehat{w} \quad \|w_{\mathrm{d}} - \widehat{w}\| \\ \text{subject to} & \widehat{w} \in \widehat{\mathscr{B}}|_{[1,T]} \quad \text{and} \quad \dim(\widehat{\mathscr{B}}) \leq \mathtt{n}. \end{array} \tag{AM $\mathscr{L}_{0,\mathtt{l}}$}$$

A solution $\widehat{\mathscr{B}}^*$ is an optimal approximate model for the signal w_{d} with bounded complexity: order at most $\mathtt{n}$.

In the general case when the data are a vector valued signal with $\mathtt{q}$ variables, the model $\mathscr{B}$ can be represented by a kernel representation, where the parameters R_i are $\mathtt{p} \times \mathtt{q}$ matrices. The block-Hankel structured low rank approximation problem is equivalent to the approximate linear time-invariant dynamic modeling problem (AM) with model class $\mathscr{M}_{\mathtt{c}_{\max}} = \mathscr{L}_{\mathtt{m},\mathtt{l}}$ and orthogonal distance fitting criterion.

Problem 2.33 (Linear time-invariant dynamic modeling) Given T samples, $\mathtt{q}$ variables, vector signal $w_{\mathrm{d}} \in (\mathbb{R}^{\mathtt{q}})^T$, a signal norm $\|\cdot\|$, and a model complexity $(\mathtt{m}, \mathtt{l})$,

$$\begin{array}{ll} \text{minimize} & \text{over } \widehat{\mathscr{B}} \text{ and } \widehat{w} \quad \|w_{\mathrm{d}} - \widehat{w}\| \\ \text{subject to} & \widehat{w} \in \widehat{\mathscr{B}}|_{[1,T]} \text{ and } \widehat{\mathscr{B}} \in \mathscr{L}^{\mathtt{q}}_{\mathtt{m},\mathtt{l}}. \end{array} \tag{AM $\mathscr{L}_{\mathtt{m},\mathtt{l}}$}$$

The solution $\widehat{\mathscr{B}}^*$ is an optimal approximate model for the signal w_{d} with complexity bounded by $(\mathtt{m}, \mathtt{l})$. Note that problem (AM $\mathscr{L}_{\mathtt{m},\mathtt{l}}$) reduces to

- (AM $\mathscr{L}_{0,\mathtt{l}}$) when $\mathtt{m} = 0$, i.e., when the model is autonomous, and
- (AM $\mathscr{L}_{\mathtt{m},0}$) when $\mathtt{l} = 0$, i.e., when the model is static.

Therefore, (AM $\mathscr{L}_{\mathtt{m},\mathtt{l}}$) is a proper generalization of linear static and dynamic autonomous data modeling problems.

Computing the optimal approximate model $\widehat{\mathscr{B}}^*$ from the solution $\widehat{p}^*$ to Problem SLRA is an exact identification problem. As in the static approximation problem, however, the parameter of a model representation is an optimization variable

of the optimization problem, used for Problem SLRA, so that a representation of the model is actually obtained directly from the optimization solver.

Similarly to the static modeling problem, the dynamic modeling problem has a maximum likelihood interpretation in the errors-in-variables setting.

Proposition 2.34 (Maximum likelihood property of an optimal dynamic model) *Assume that the data w_{d} are generated in the errors-in-variables setting*

$$w_{\mathrm{d}} = w_0 + \widetilde{w}, \quad \text{where } w_0 \in \mathscr{B}_0|_{[1,T]} \in \mathscr{L}^{\mathrm{q}}_{\mathrm{m},1} \text{ and } \widetilde{w} \sim \mathrm{N}(0, vI). \tag{EIV}$$

Then an optimal approximate model $\widehat{\mathscr{B}}^*$, solving (AM $\mathscr{L}_{\mathrm{m},1}$) *with $\|\cdot\| = \|\cdot\|_2$ is a maximum likelihood estimator for the true model $\mathscr{B}_0$.*

The proof is analogous to the proof of Proposition 2.32 and is skipped.

In Chap. 3, we describe local optimization methods for general affinely structured low rank approximation problems and show how in the case of Hankel, Toeplitz, and Sylvester structured problem, the matrix structure can be exploited for efficient cost function evaluation.

2.6 Notes

The concept of the most powerful unfalsified model is introduced in Willems (1986, Definition 4). See also Antoulas and Willems (1993), Kuijper and Willems (1997), Kuijper (1997) and Willems (1997). Kung's method for approximate system realization is presented in Kung (1978).

Modeling by the orthogonal distance fitting criterion (misfit approach) is initiated in Willems (1987) and further on developed in Roorda and Heij (1995), Roorda (1995a, 1995b), Markovsky et al. (2005b), where algorithms for solving the problems are developed. A proposal for combination of misfit and latency for linear time-invariant system identification is made in Lemmerling and De Moor (2001).

Weighted low rank approximation methods are developed in Gabriel and Zamir (1979), De Moor (1993), Wentzell et al. (1997), Markovsky et al. (2005a), Manton et al. (2003), Srebro (2004), Markovsky and Van Huffel (2007). The analytic solution of circulant structured low rank approximation problem is derived independently in the optimization community (Beck and Ben-Tal 2006) and in the systems and control community (Vanluyten et al. 2005).

Equivalence of Low Rank Approximation and Principal Component Analysis

The principal component analysis method for dimensionality reduction is usually introduced in a stochastic setting as maximization of the variance of the projected

data on a subspace. Computationally, however, the problem of finding the principal components and the corresponding principal vectors is an eigenvalue/eigenvector decomposition problem for the sample covariance matrix

$$\Psi(\mathscr{D}) := \Phi(\mathscr{D})\Phi^{\top}(\mathscr{D}), \quad \text{where } \Phi(\mathscr{D}) := \begin{bmatrix} d_1 \cdots d_N \end{bmatrix}.$$

From this algorithmic point view, the equivalence of principal component analysis and low rank approximation problem is a basic linear algebra fact: the space spanned by the first $\mathtt{m}$ principal vectors of $\mathscr{D}$ coincides with the model $\widehat{\mathscr{B}} = \text{image}(\Phi(\widehat{\mathscr{D}}))$, where $\mathscr{D}$ is a solution of the low rank approximation problem (LRA).

References

Abelson H, diSessa A (1986) Turtle geometry. MIT Press, New York

Antoulas A, Willems JC (1993) A behavioral approach to linear exact modeling. IEEE Trans Automat Control 38(12):1776–1802

Beck A, Ben-Tal A (2006) A global solution for the structured total least squares problem with block circulant matrices. SIAM J Matrix Anal Appl 27(1):238–255

De Moor B (1993) Structured total least squares and L_2 approximation problems. Linear Algebra Appl 188–189:163–207

Gabriel K, Zamir S (1979) Lower rank approximation of matrices by least squares with any choice of weights. Technometrics 21:489–498

Kuijper M (1997) An algorithm for constructing a minimal partial realization in the multivariable case. Control Lett 31(4):225–233

Kuijper M, Willems JC (1997) On constructing a shortest linear recurrence relation. IEEE Trans Automat Control 42(11):1554–1558

Kung S (1978) A new identification method and model reduction algorithm via singular value decomposition. In: Proc 12th asilomar conf circuits, systems, computers, Pacific Grove, pp 705–714

Lemmerling P, De Moor B (2001) Misfit versus latency. Automatica 37:2057–2067

Manton J, Mahony R, Hua Y (2003) The geometry of weighted low-rank approximations. IEEE Trans Signal Process 51(2):500–514

Markovsky I, Van Huffel S (2007) Left vs right representations for solving weighted low rank approximation problems. Linear Algebra Appl 422:540–552

Markovsky I, Rastello ML, Premoli A, Kukush A, Van Huffel S (2005a) The element-wise weighted total least squares problem. Comput Stat Data Anal 50(1):181–209

Markovsky I, Willems JC, Van Huffel S, Moor BD, Pintelon R (2005b) Application of structured total least squares for system identification and model reduction. IEEE Trans Automat Control 50(10):1490–1500

Roorda B (1995a) Algorithms for global total least squares modelling of finite multivariable time series. Automatica 31(3):391–404

Roorda B (1995b) Global total least squares—a method for the construction of open approximate models from vector time series. PhD thesis, Tinbergen Institute

Roorda B, Heij C (1995) Global total least squares modeling of multivariate time series. IEEE Trans Automat Control 40(1):50–63

Srebro N (2004) Learning with matrix factorizations. PhD thesis, MIT

Vanluyten B, Willems JC, De Moor B (2005) Model reduction of systems with symmetries. In: Proc 44th IEEE conf dec control, Seville, Spain, pp 826–831

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

Willems JC (1986) From time series to linear system—Part II. Exact modelling. Automatica 22(6):675–694
Willems JC (1987) From time series to linear system—Part III. Approximate modelling. Automatica 23(1):87–115
Willems JC (1997) On interconnections, control, and feedback. IEEE Trans Automat Control 42:326–339

http://www.springer.com/978-1-4471-2226-5

Low Rank Approximation
Algorithms, Implementation, Applications
Markovsky, I.
2012, X, 258 p., Hardcover
ISBN: 978-1-4471-2226-5