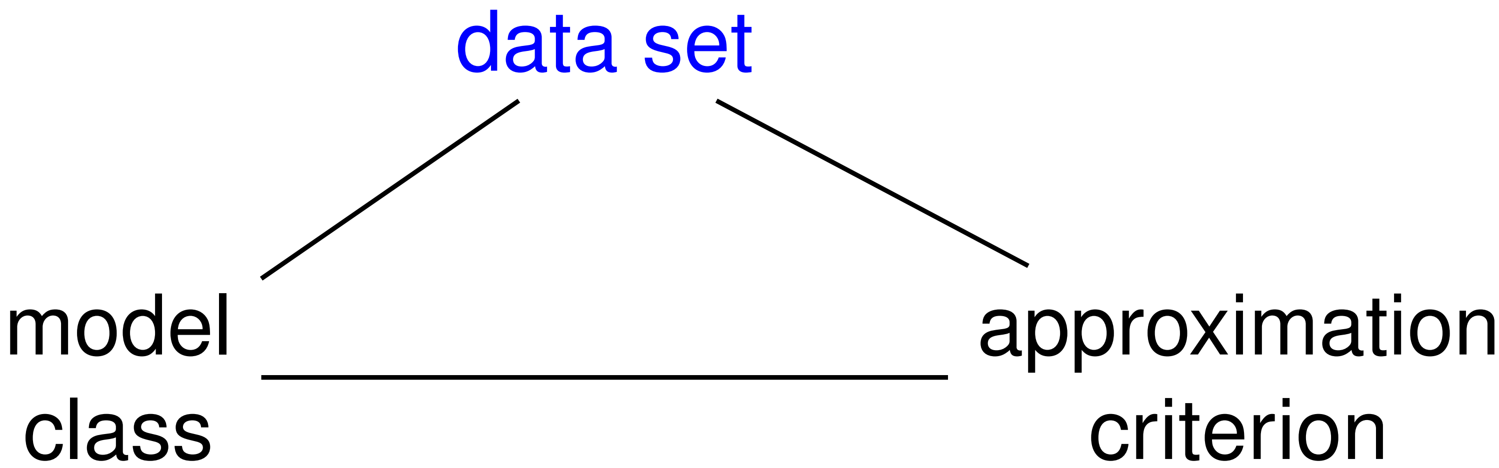
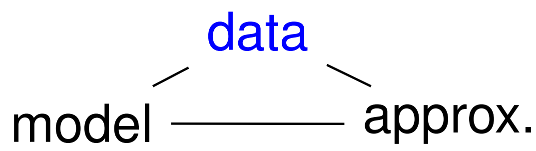
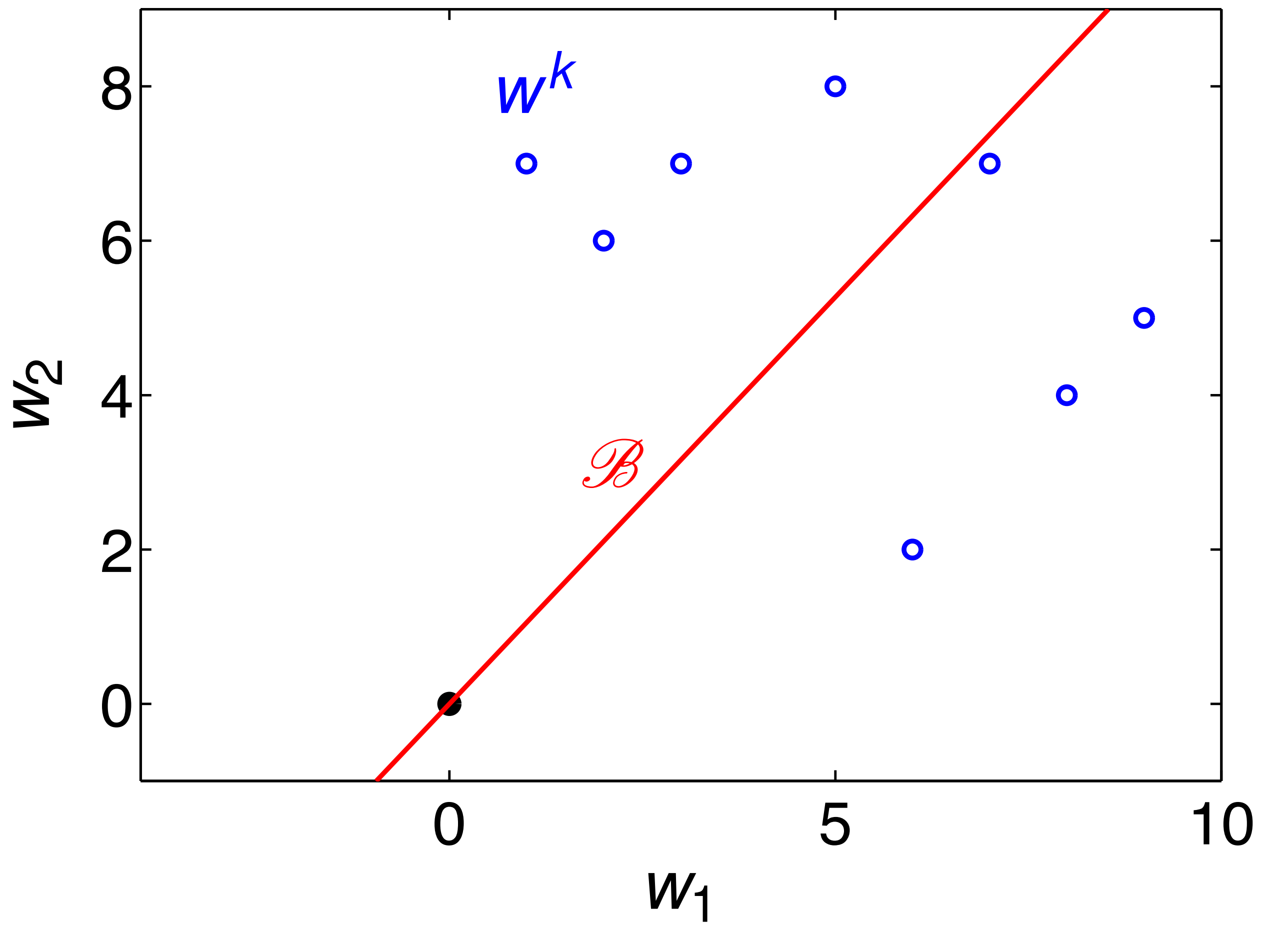
First is the data . . .



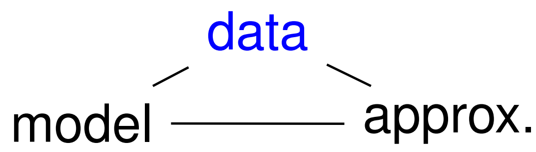
Line fitting (linear static model)



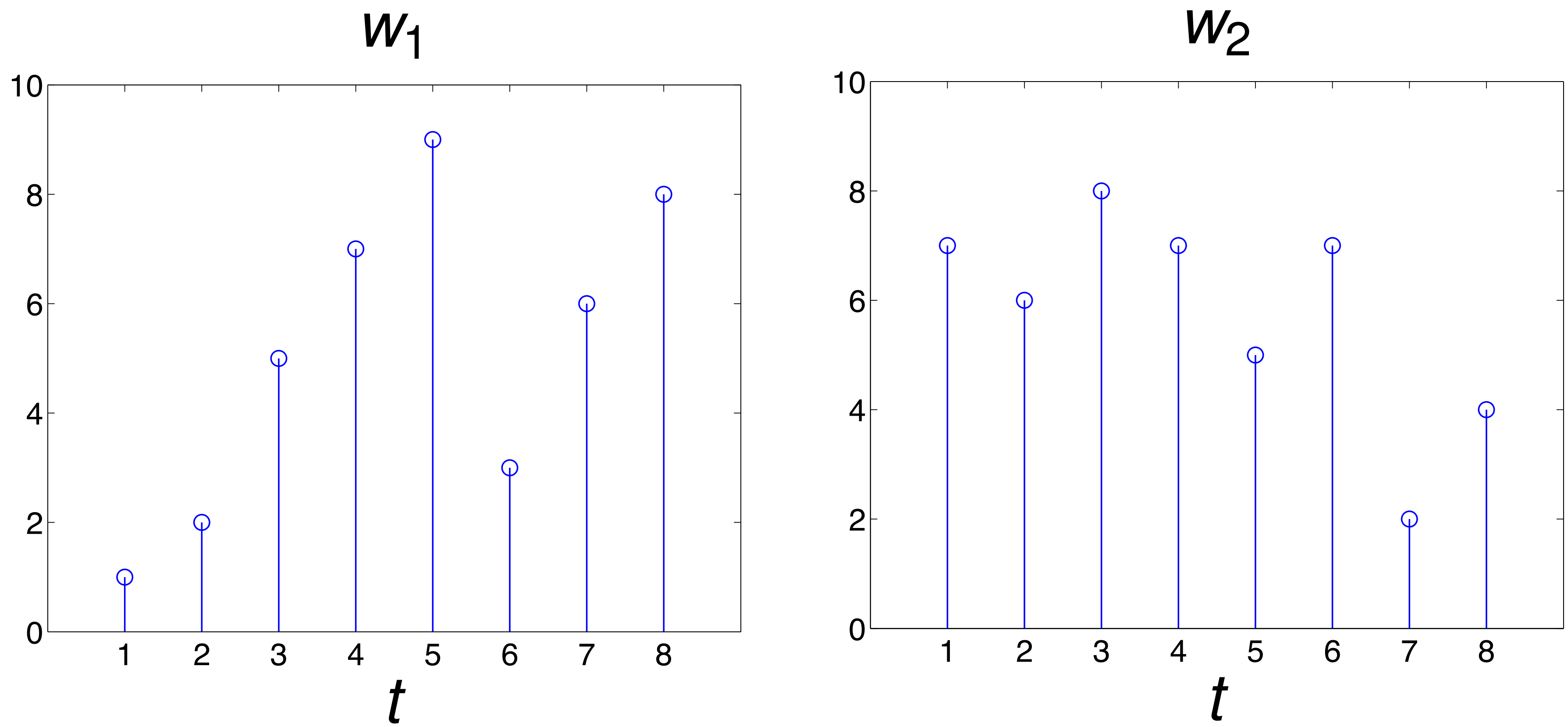
$w^1, \ldots, w^N$ — data points (the order is not important)



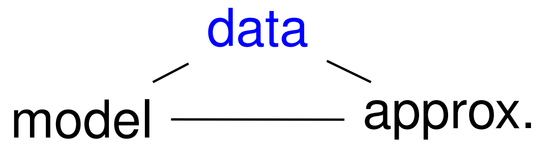
Time series data (dynamic model)



$w(1), \ldots, w(T)$ — samples in time (the order is important)



Summary: data



- the data is a set $w = \{ w^1, \dots, w^N \}$

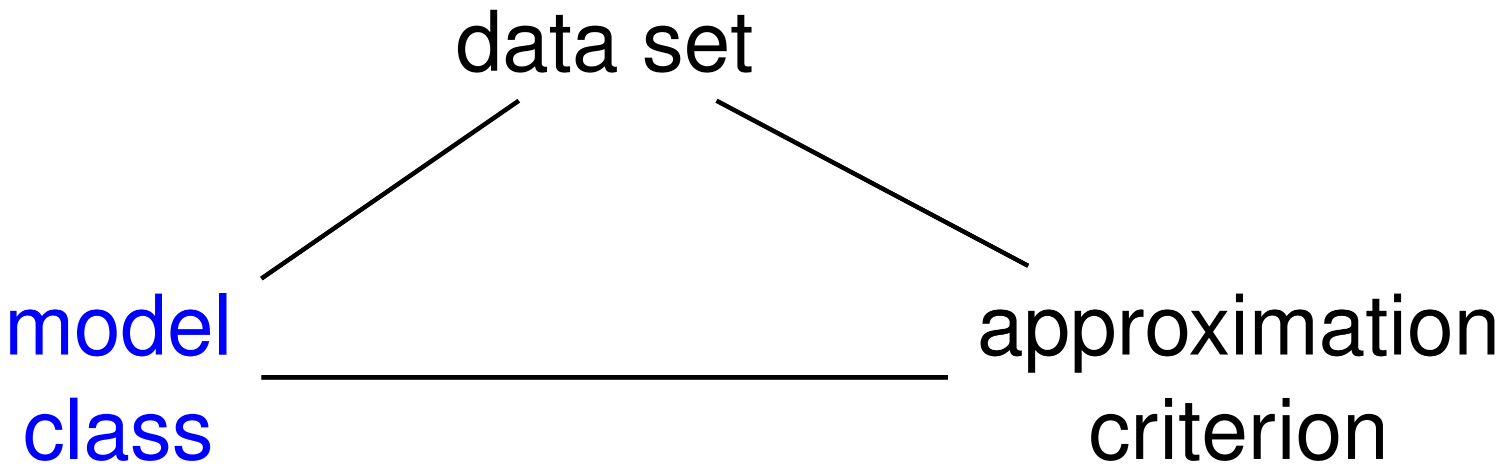
- of vector valued $w^k = \begin{bmatrix} w_1^k \\ \vdots \\ w_q^k \end{bmatrix}$

- time series $w_i^k = \big(w_i^k(1), \dots, w_i^k(T_k) \big)$

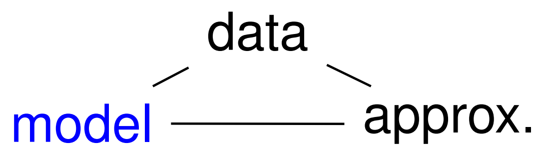
N	—	# of repeated experiments
q	—	# of variables
T_k	—	# of time samples in the *k*th exp.

- in static problems, $T_1 = \cdots = T_N = 1$

Next is the model class . . .

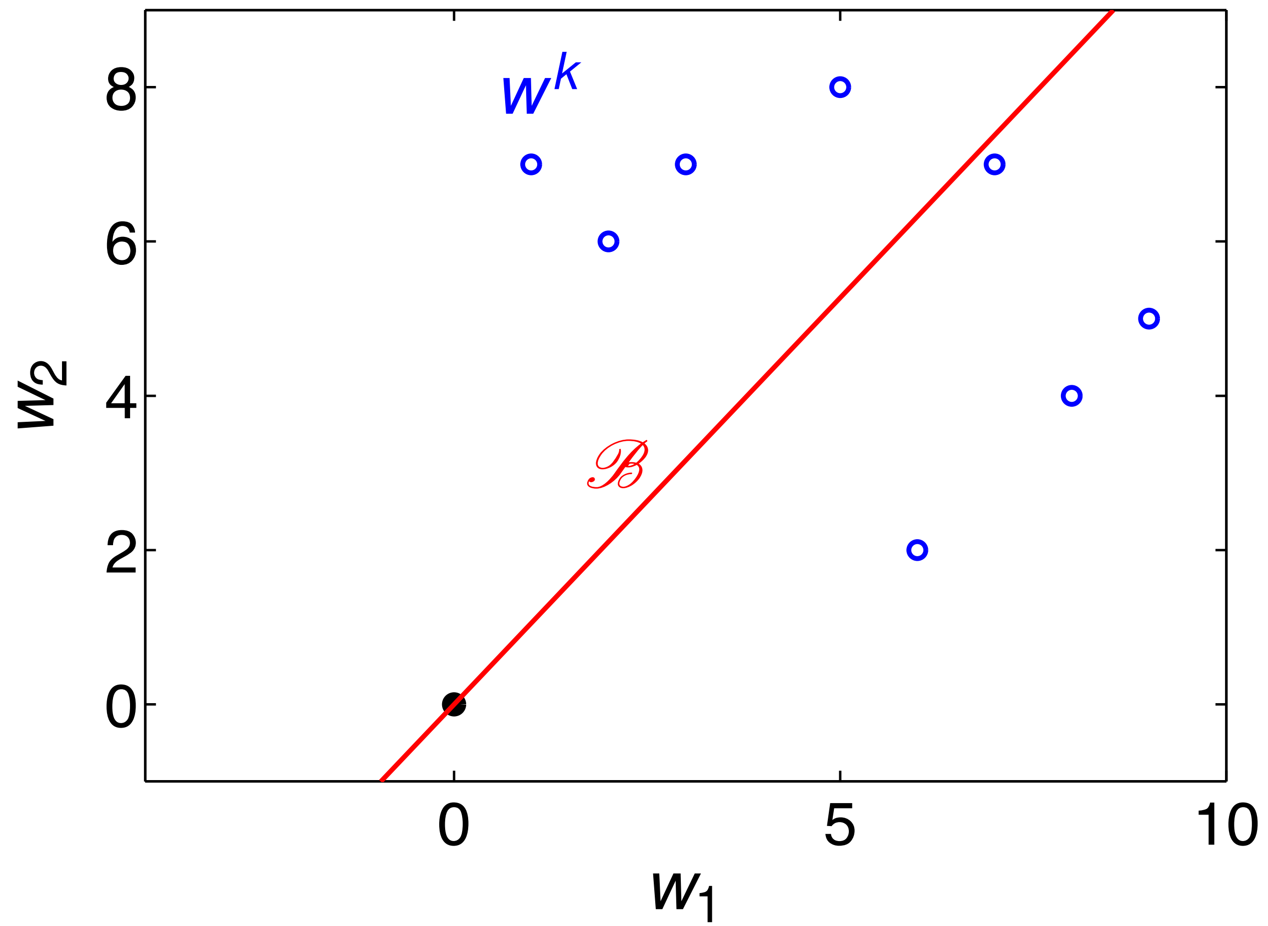


Line fitting (linear static model)

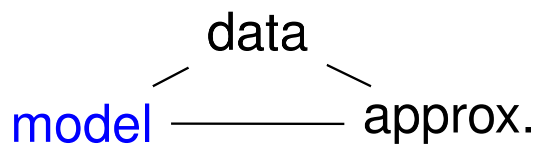


$\mathscr{B}$ — model: line through the origin

$\mathscr{M}$ — model class: all lines through the origin

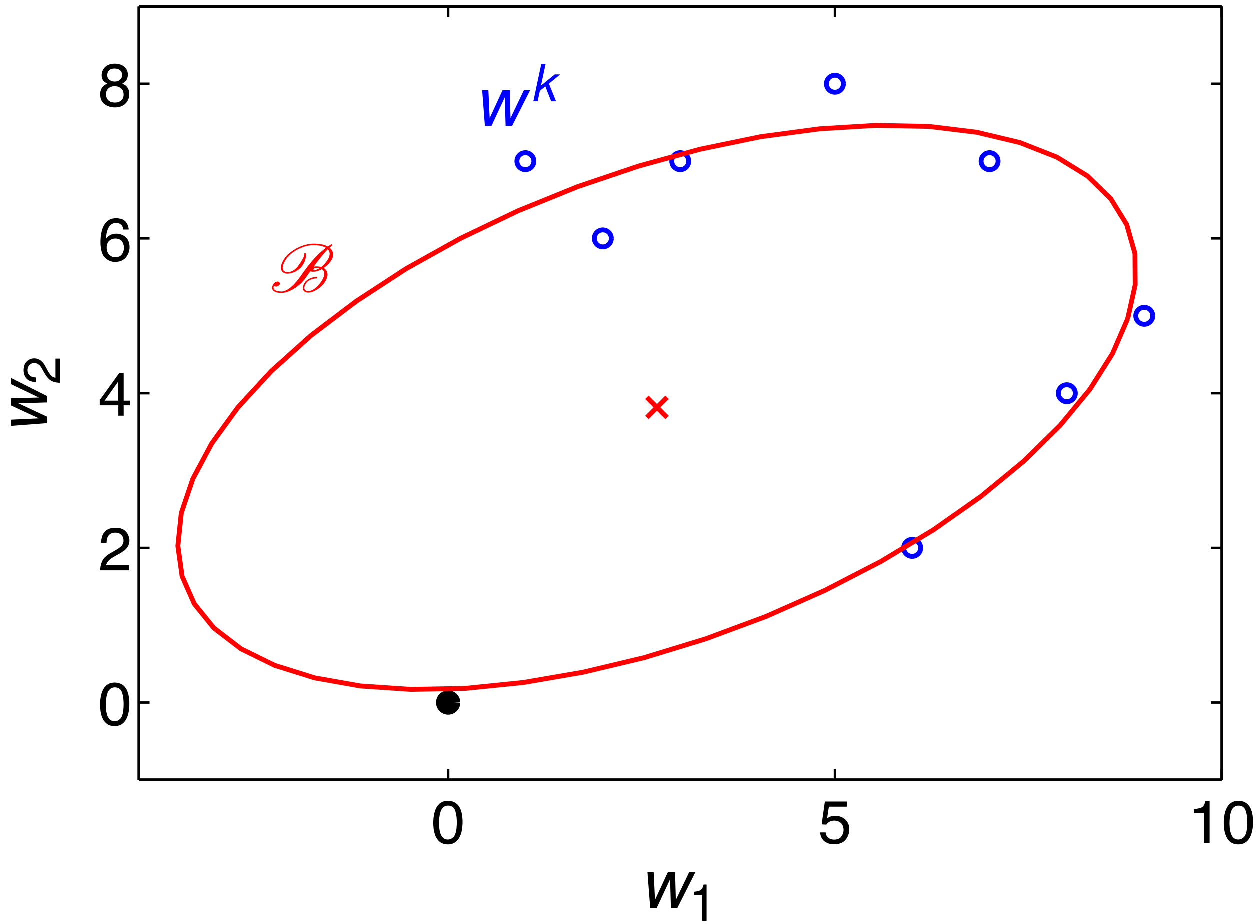


Conic section fitting (quadratic static model)



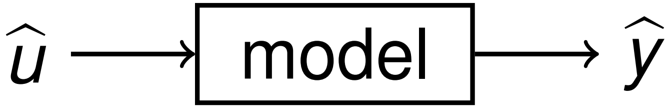
$\mathscr{B}$ — model: conic section

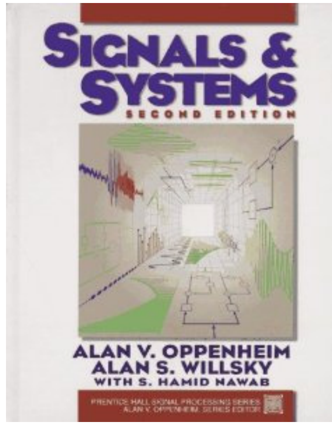
$\mathscr{M}$ — model class: all conic sections



Classical definition of dynamical model

- dynamical model is signal processor

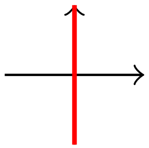
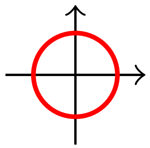




- specified by a map $\widehat{y} = f(\widehat{u})$

- "state space model", "transfer function model", ...

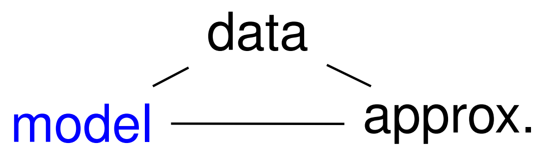
- however, lines and conic sections may not be graphs

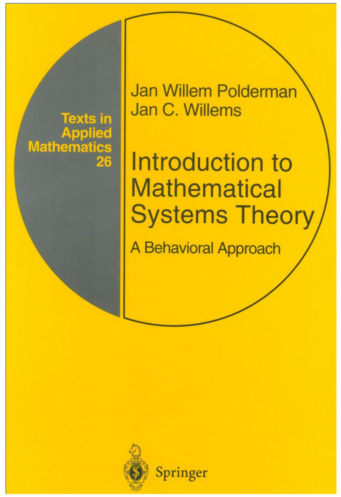
- *e.g.*,  ,  can't be represented by $f : \widehat{u} \mapsto \widehat{y}$

"good definition should formalize sensible intuition"

Jan Willems, Paradigms and puzzles, TAC'91

Behavioral definition of model





- a model is a subset

$$\mathscr{B} = \{\, \widehat{w} \mid g(\widehat{w}) = 0 \text{ holds} \,\}$$

- represented by an implicit function g

- in the static case, $g(\widehat{w}) = 0$ is algebraic equation

- in the dynamic case, $g(\widehat{w}) = 0$ is difference equation

- $\widehat{w} = \begin{bmatrix} \widehat{u} \\ \widehat{y} \end{bmatrix}$, $\widehat{y} = f(\widehat{u})$ is a special case of $g(\widehat{w}) = 0$

$$(g(\widehat{u}, \widehat{y}) = \widehat{y} - f(\widehat{u}))$$

Summary: model

data
model — approx.

- three data modeling examples:

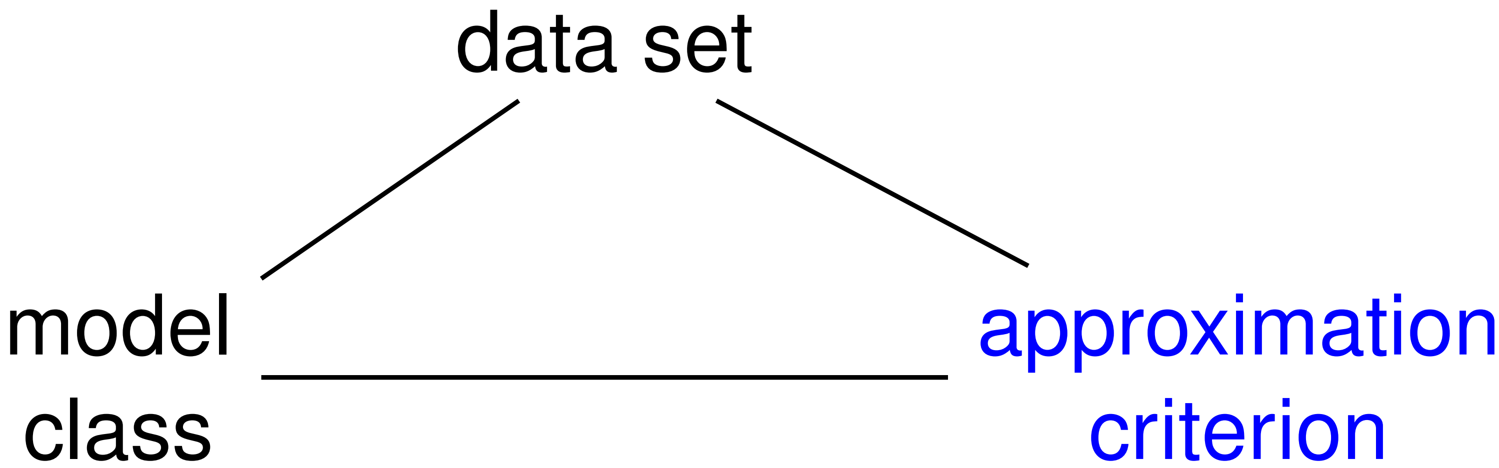
problem	model
line fitting	static linear
conic section fitting	static nonlinear
system identification	dynamic

- two definitions of a model:

classical	behavioral
map $\widehat{y} = f(\widehat{u})$	set $\{\, \widehat{w} \mid g(\widehat{w}) = 0 \,\}$
f — function	g — relation

- the classical one can not deal with all examples

Finally, the approximation criterion . . .

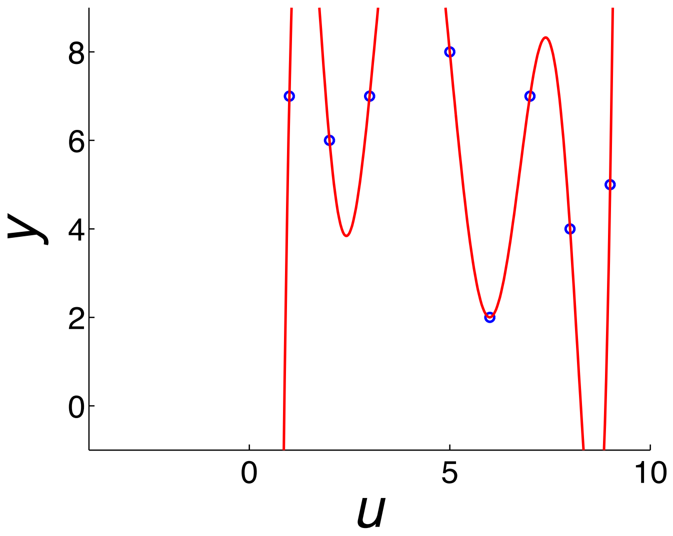


Exact model

$$w \subset \mathscr{B} \quad \iff \quad w^1, \ldots, w^N \in \mathscr{B}$$

$$\iff: \quad \text{"}w \text{ is exact data of } \mathscr{B}\text{"}$$

- two well known exact modeling problems
  - realization: LTI model class, impulse resp. data
  - interpolation: static nonlinear model class



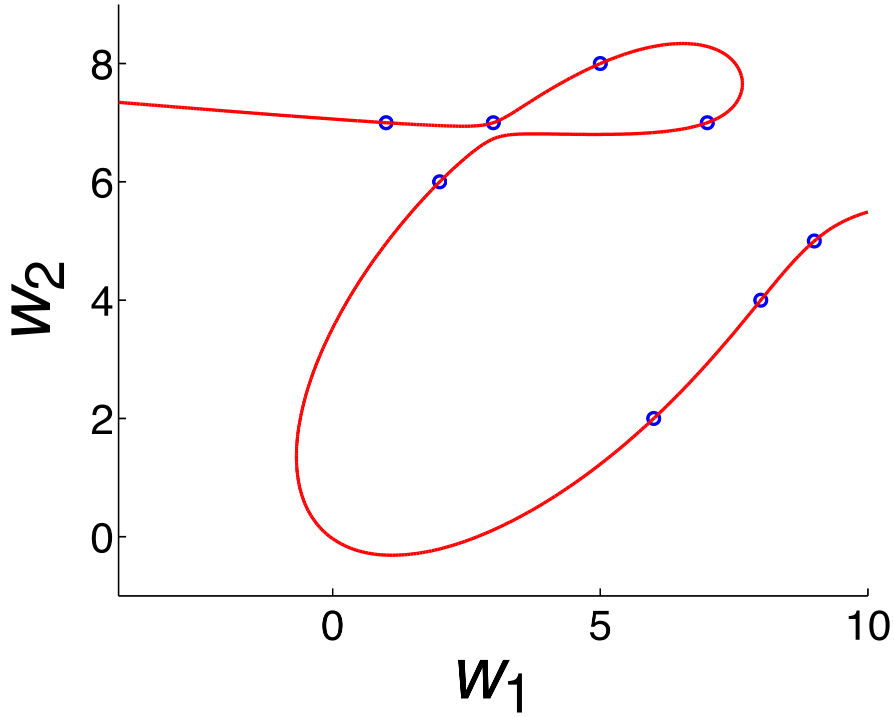
$$\mathscr{B} = \left\{ \begin{bmatrix} \widehat{u} \\ \widehat{y} \end{bmatrix} \mid \widehat{y} = f(\widehat{u}) \right\}$$

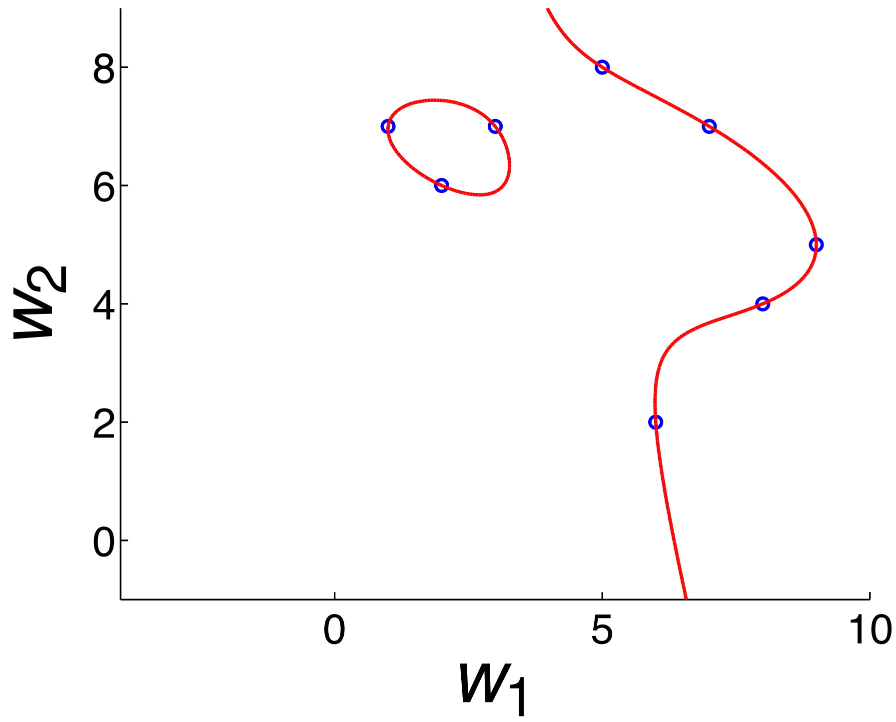
f is 8th order polynomial

Exact 3rd order nonlinear static models

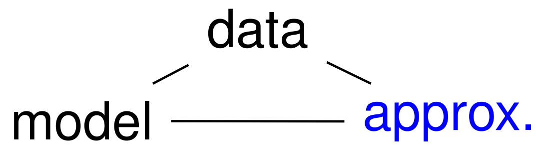
$$\mathscr{B} = \left\{ \begin{bmatrix} \widehat{w}_1 \\ \widehat{w}_2 \end{bmatrix} \mid g(\widehat{w}_1, \widehat{w}_2) = 0 \right\}$$

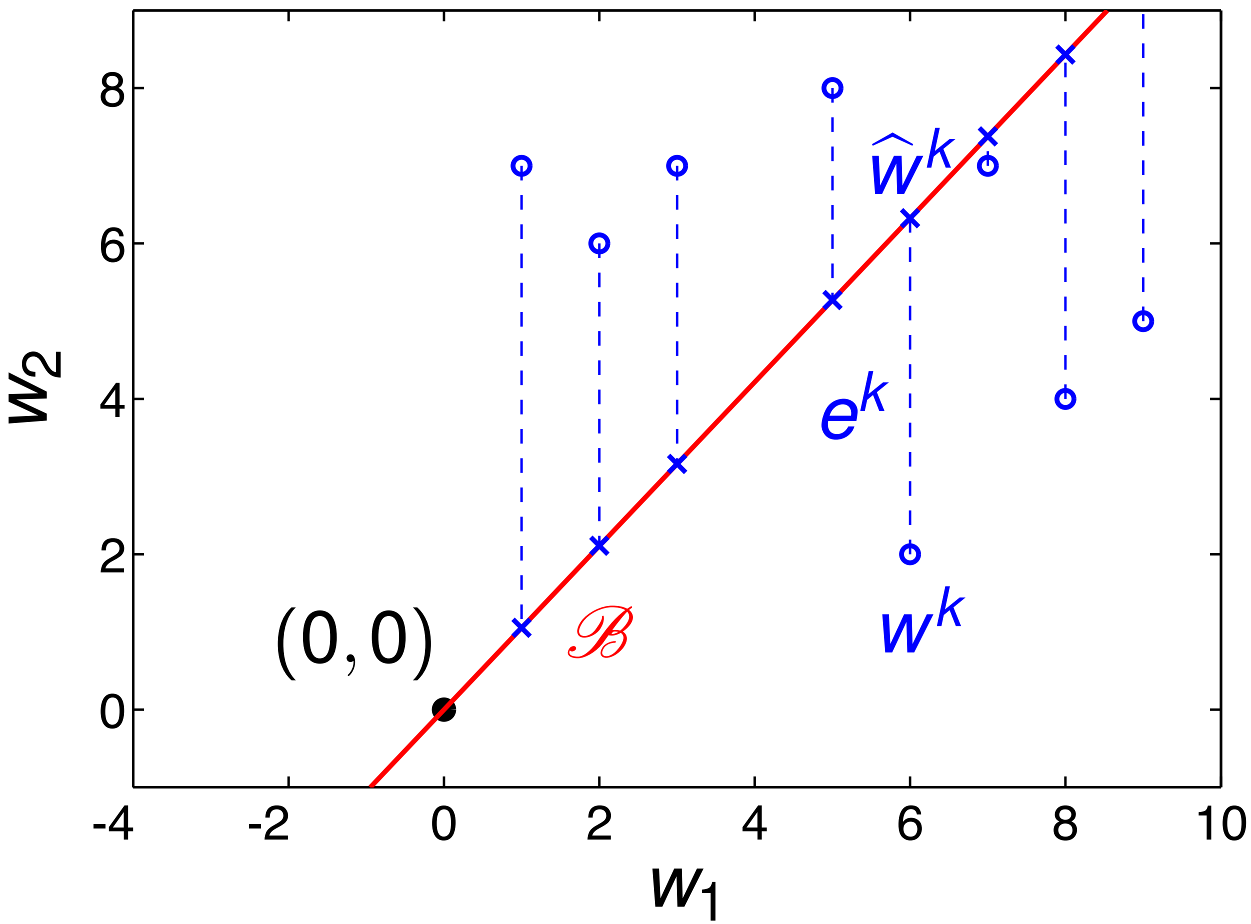
g is 3rd order polynomial in $\widehat{w}_1$, $\widehat{w}_2$





Ordinary least squares

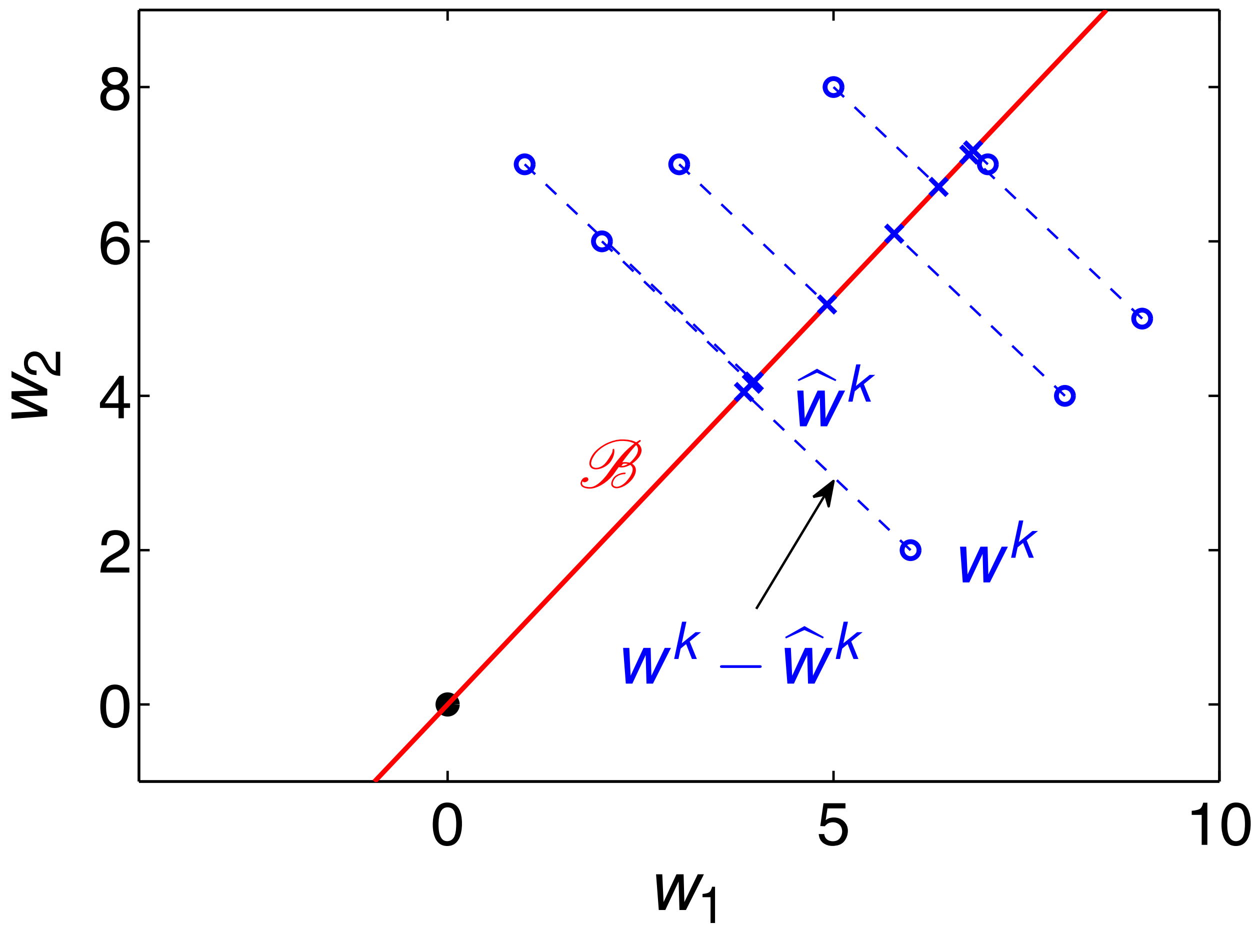


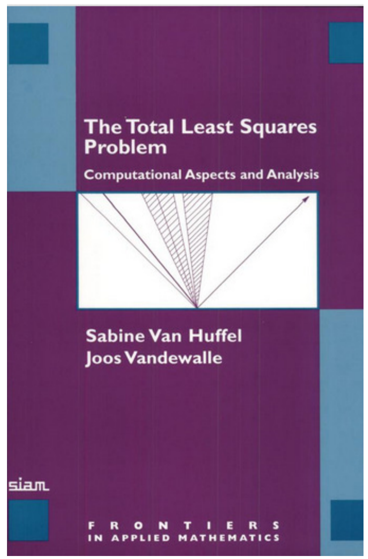


Total least squares

data

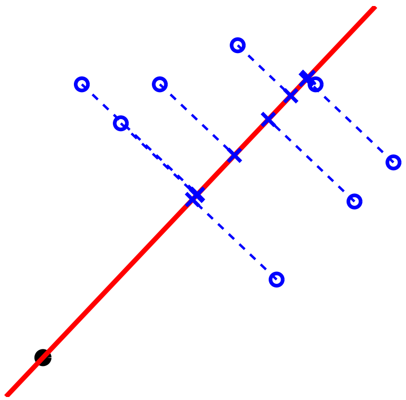
model — approx.





Linear static case

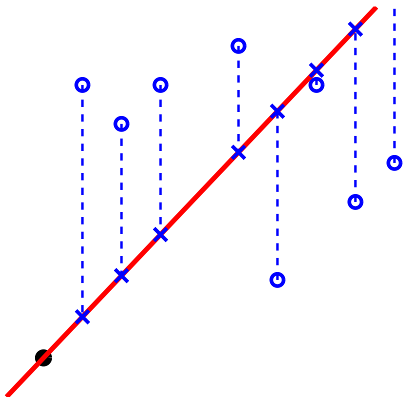
data
model ——— approx.



- total least squares

$$\min_{\widehat{u},\widehat{y},\theta} \left\| \begin{bmatrix} u-\widehat{u} & y-\widehat{y} \end{bmatrix} \right\|_{\mathrm{F}} \text{ s.t. } \underbrace{\widehat{u}\theta = \widehat{y}}_{(\widehat{u},\widehat{y})\subset\mathscr{B}(\theta)}$$

$\widehat{w} = (\widehat{u},\widehat{y})$ approximates $w = (u,y)$



- ordinary least squares

$$\min_{\widehat{e},\theta} \|\widehat{e}\|_2 \quad \text{s.t.} \quad \underbrace{u\theta = y + \widehat{e}}_{(\widehat{e},u,y)\subset\mathscr{B}_{\text{ext}}(\theta)}$$

$\widehat{e}$ is unobserved (latent) input

Exact models in the approximation criteria

- Misfit approach:

  modify w as little as possible,
  so that $\widehat{w}$ is exact

  $\|w - \widehat{w}\|$ is the misfit criterion

- Latency approach:

  augment $\mathscr{B}$ by as small as possible e,
  so that (e, w) is exact

  $\|e\|$ is the latency criterion

Deterministic vs stochastic setting

data

model — approx.

- stochastic estimation $\leftrightarrow$ deterministic approx.

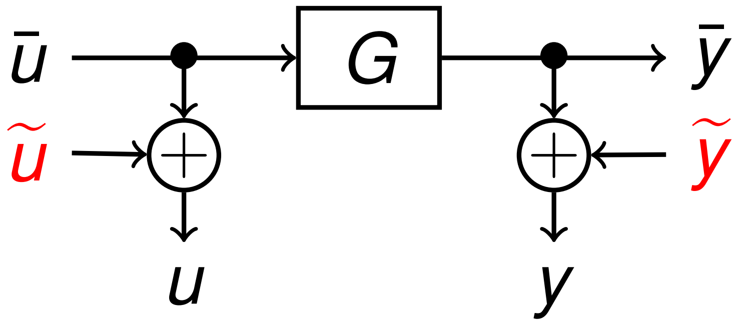
noise model $\xleftrightarrow{\text{maximum likelihood}}$ cost function

- also in control

LQG control $\leftrightarrow$ H_2 optimal control

Misfit and latency in the stochastic setting

EIV $\leftrightarrow$ misfit

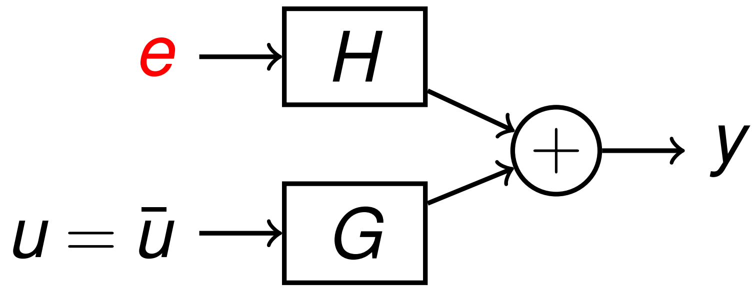


$\widetilde{u}, \widetilde{y}$ — measurement errors

$$\min_{\widehat{w} \subset \mathscr{B}} \|w - \widehat{w}\|$$

$$\mathscr{B} := \left\{ \begin{bmatrix} \widehat{u} \\ \widehat{y} \end{bmatrix} \mid \widehat{y} = \widehat{G}\widehat{u} \right\}$$

ARMAX $\leftrightarrow$ latency

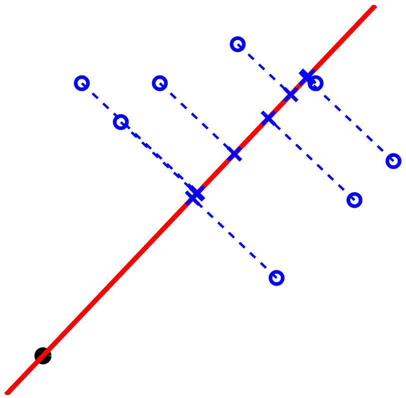


e — disturbance

$$\min_{(\widehat{e}, w) \subset \mathscr{B}_{\text{ext}}} \|\widehat{e}\|$$

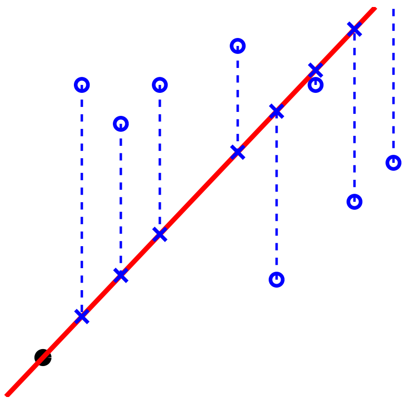
$$\mathscr{B}_{\text{ext}} := \left\{ \begin{bmatrix} \widehat{e} \\ u \\ y \end{bmatrix} \mid y = [\widehat{H} \ \widehat{G}] \begin{bmatrix} \widehat{e} \\ u \end{bmatrix} \right\}$$

Summary: approximation criterion



- TLS $\leftrightarrow$ misfit $\leftrightarrow$ errors-in-variables

$$\min_{\widehat{w}\subset\mathscr{B}} \|w-\widehat{w}\| \quad \left(\begin{array}{c}\text{projection}\\ \text{of } w \text{ on } \mathscr{B}\end{array}\right)$$



- OLS $\leftrightarrow$ latency $\leftrightarrow$ ARMAX

$$\min_{(\widehat{e},w)\in\mathscr{B}_{\text{ext}}} \|\widehat{e}\|$$

A general problem

$$\underset{w}{\text{data}} \xrightarrow{\text{identification}} \underset{\mathscr{B} \in \mathscr{M}}{\text{model}}$$

the aim is to obtain "simple" and "accurate" model:

"accurate" $\rightarrow$ min. error$(w, \widehat{\mathscr{B}}) =$ misfit/latency

"simple" $\rightarrow$ Occam's razor principle:
among equally accurate models,
choose the simplest

Model complexity

- simple models are small models

$$\mathscr{B}_1 \subset \mathscr{B}_2 \quad \Longrightarrow \quad \mathscr{B}_1 \text{ is simpler than } \mathscr{B}_2$$

- nonlinear model complexity is an open problem
- in the linear time-invariant case, $\mathscr{B}$ is a subspace

$$\text{size of the model} = \text{dimension of } \mathscr{B}$$

- however, models with inputs are infinite dimensional

Linear time-invariant model's complexity

- restriction of $\mathscr{B}$ on an interval $[1, T]$

$$\mathscr{B}|_T = \{\, w = \big(w(1), \dots, w(T)\big) \mid \exists\, w_{\mathrm{p}}, w_{\mathrm{f}}, \text{ such that } (w_{\mathrm{p}}, w, w_{\mathrm{f}}) \in \mathscr{B} \,\}$$

- for sufficiently large T

$$\dim(\mathscr{B}|_T) = (\text{\# of inputs}) \cdot T + (\text{order})$$

$$\text{complexity}(\mathscr{B}) = \begin{bmatrix} \mathrm{m} \\ \ell \end{bmatrix} \begin{matrix} \to & \text{\# of inputs} \\ \to & \text{order or lag} \end{matrix}$$

- $\mathscr{L}_{\mathrm{m},\ell}$ — set of LTI systems of bounded complexity

Complexity selection

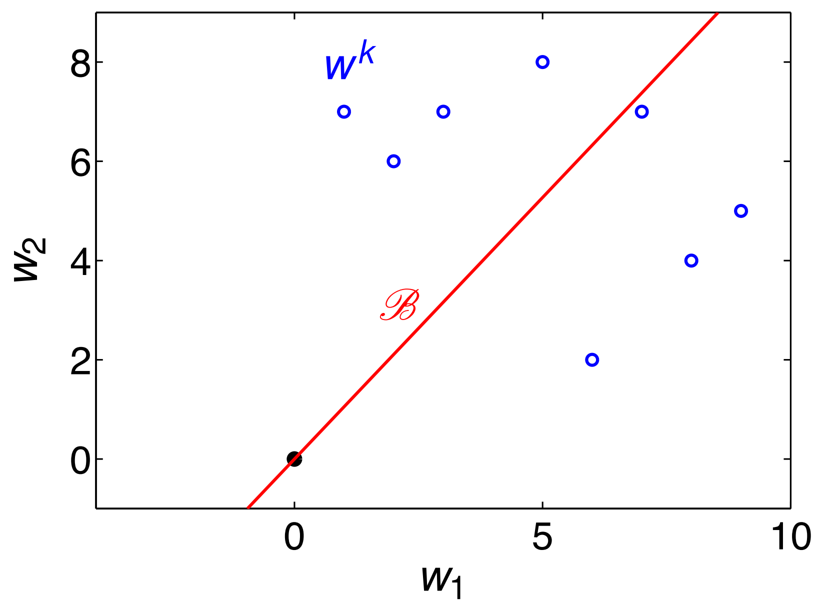
- if m is given and fixed, choosing the complexity is an

  order selection problem

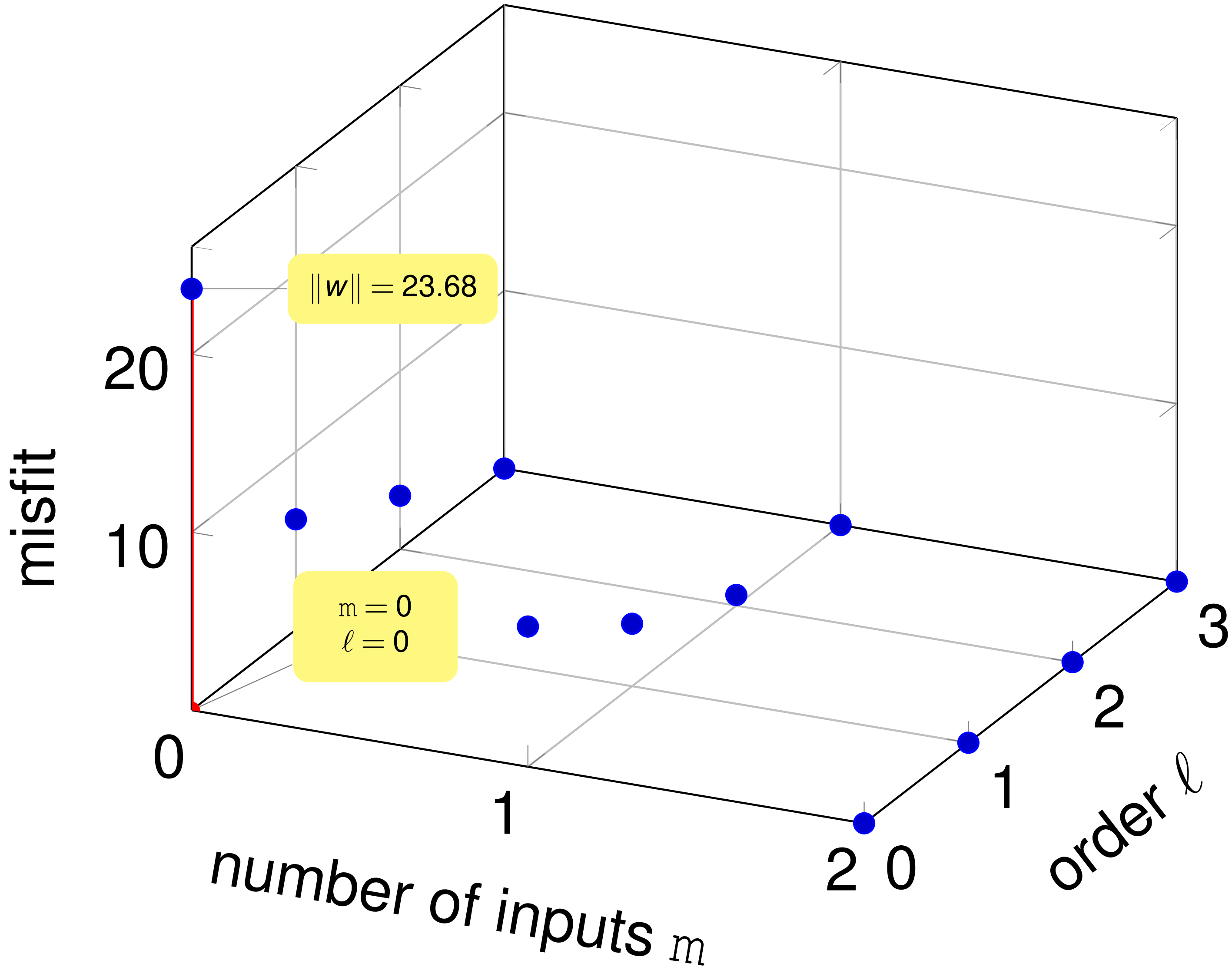
- in general, choosing the complexity involves

  order selection and input selection

  illustrated next on the example from the introduction

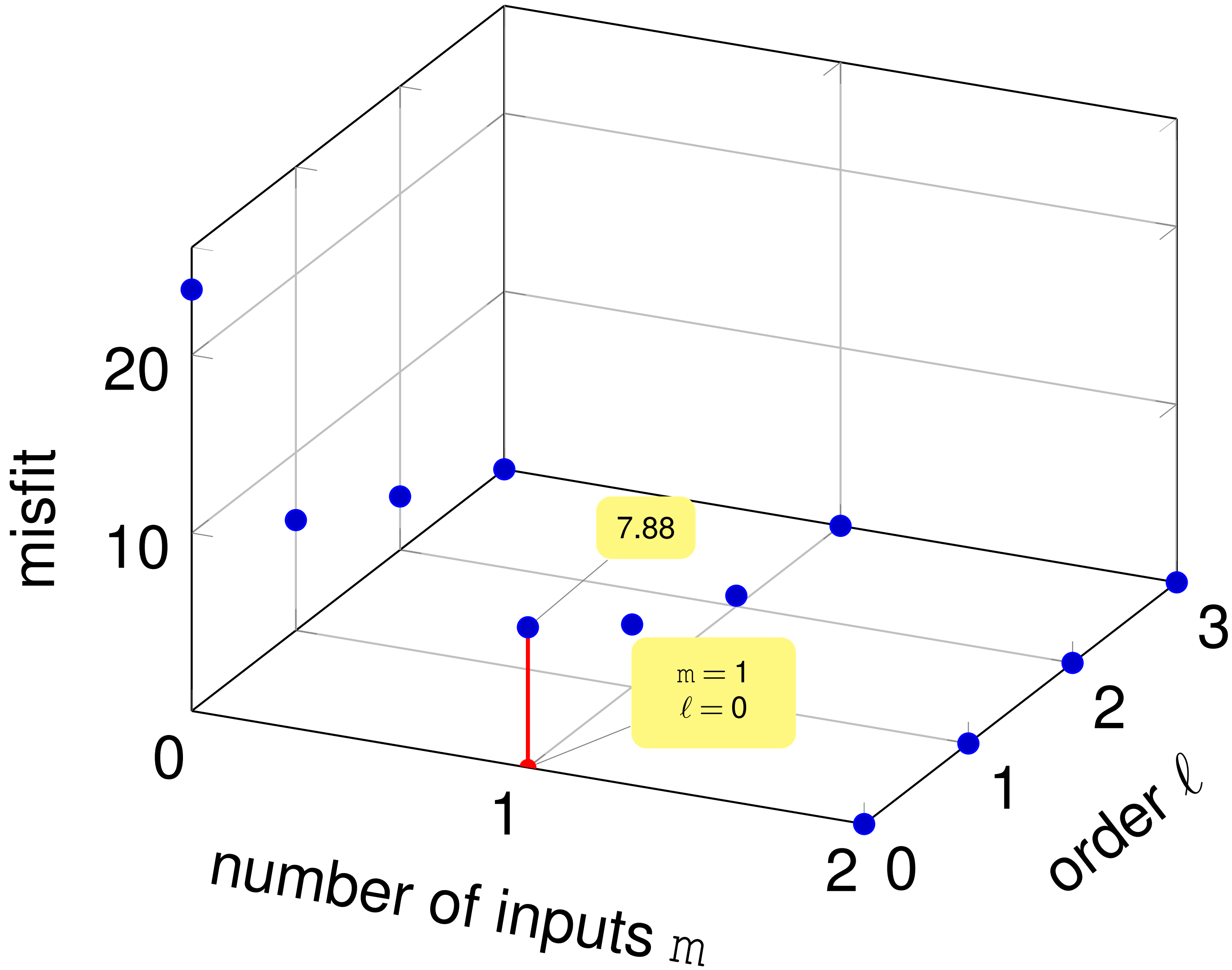


Example: misfit-complexity trade-off



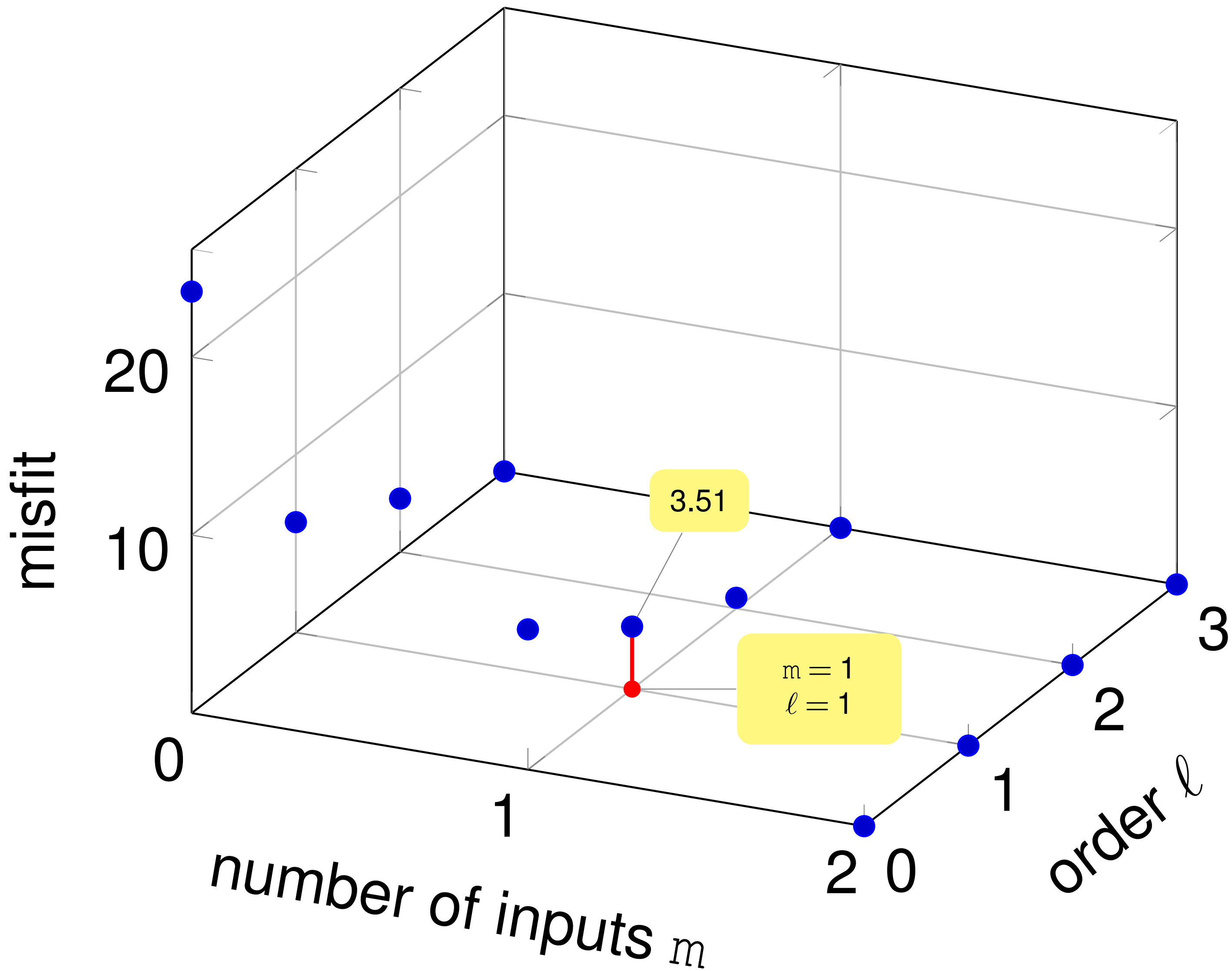
$m = 0$, $\ell = 0 \implies \mathscr{B} = \{0\}$ is the only model

Example: misfit-complexity trade-off



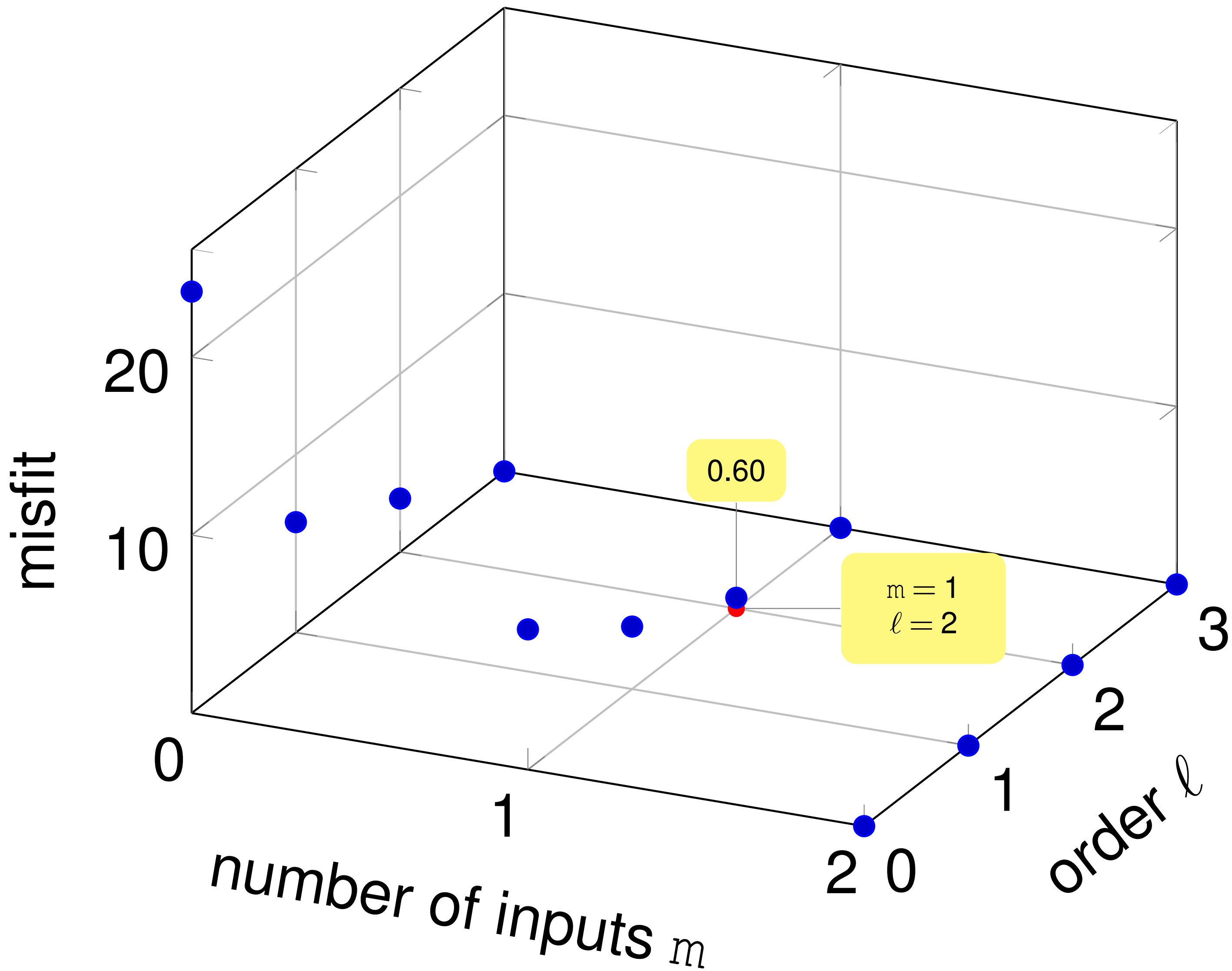
$\mathtt{m} = 1,\ \ell = 0 \implies \mathscr{B}$ is a line through 0

Example: misfit-complexity trade-off



$\mathtt{m} = 1$, $\ell = 1 \implies \mathscr{B}$ is 1st order SISO

Example: misfit-complexity trade-off



$\mathtt{m} = 1$, $\ell = 2 \implies \mathscr{B}$ is 2nd order SISO

Approximation error-complexity trade-off

$$\text{minimize} \quad \text{over } \widehat{\mathscr{B}} \in \mathscr{L} \quad \begin{bmatrix} \text{error}(w, \widehat{\mathscr{B}}) \\ \text{complexity}(\widehat{\mathscr{B}}) \end{bmatrix}$$

three ways to "scalarize" the problem:

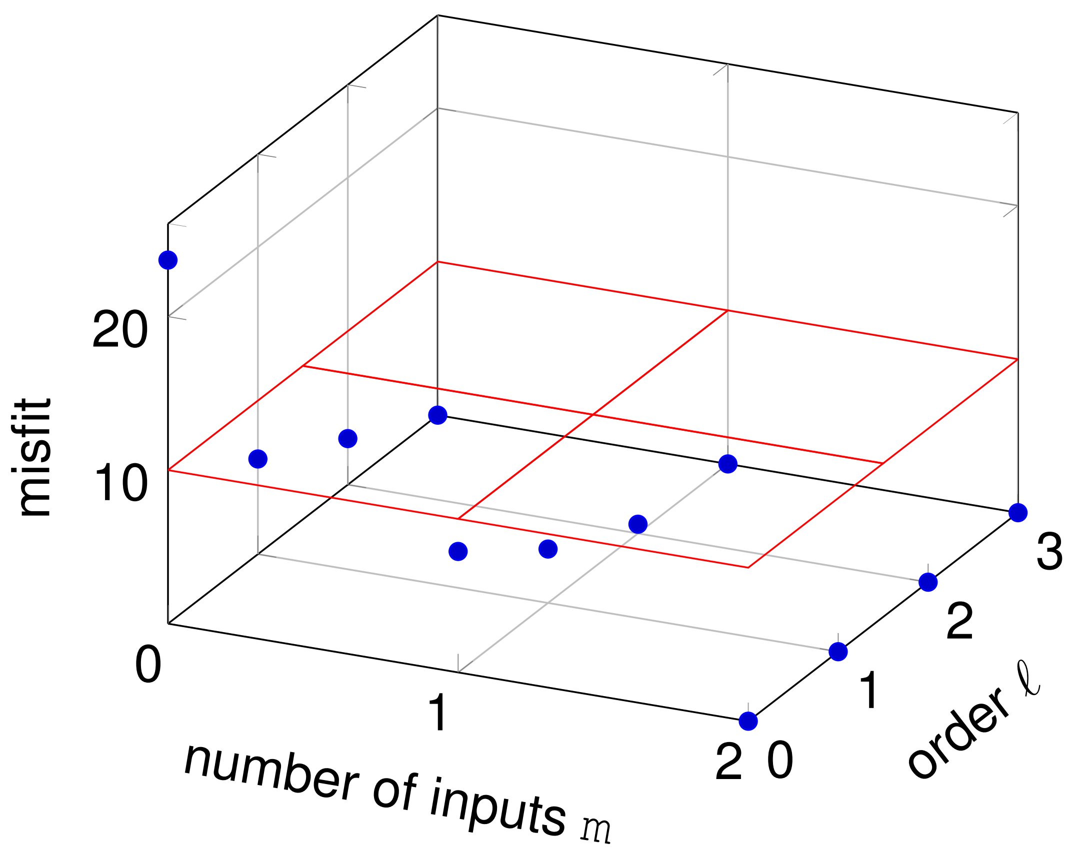
1. $\text{minimize over } \widehat{\mathscr{B}} \in \mathscr{L} \ \text{error}(w, \widehat{\mathscr{B}}) + \lambda \text{complexity}(\widehat{\mathscr{B}})$

2. $\begin{array}{ll} \text{minimize} & \text{over } \widehat{\mathscr{B}} \in \mathscr{L} \quad \text{complexity}(\widehat{\mathscr{B}}) \\ \text{subject to} & \text{error}(w, \widehat{\mathscr{B}}) \leq \mu \end{array}$

3. $\begin{array}{ll} \text{minimize} & \text{over } \widehat{\mathscr{B}} \quad \text{error}(w, \widehat{\mathscr{B}}) \\ \text{subject to} & \widehat{\mathscr{B}} \in \mathscr{L}_{\mathrm{m},\ell} \end{array}$

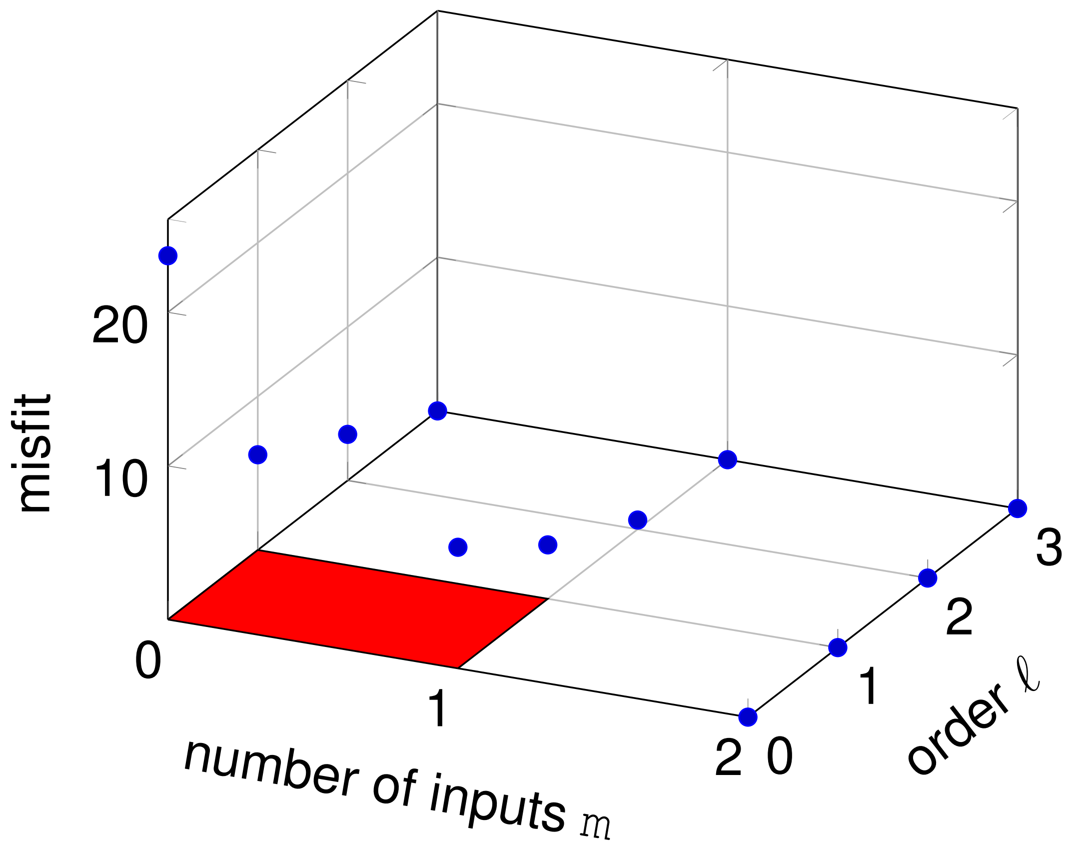
Complexity minimization with error bound

$$\text{minimize} \quad \text{over } \widehat{\mathscr{B}} \in \mathscr{L} \quad \text{complexity}(\widehat{\mathscr{B}})$$
$$\text{subject to} \quad \text{error}(w, \widehat{\mathscr{B}}) \leq \mu$$



Error minimization with complexity bound

$$\begin{aligned} &\text{minimize} \quad \text{over } \widehat{\mathscr{B}} \quad \text{error}(w, \widehat{\mathscr{B}}) \\ &\text{subject to} \quad \widehat{\mathscr{B}} \in \mathscr{L}_{\mathtt{m},\ell} \end{aligned}$$



Summary: error-complexity trade-off

- LTI model complexity

$$\text{complexity}(\mathscr{B}) = \begin{bmatrix} \mathtt{m} \\ \ell \end{bmatrix} \begin{array}{l} \rightarrow \quad \text{\# of inputs} \\ \rightarrow \quad \text{order or lag} \end{array}$$

- error-complexity trade-off

$$\text{minimize} \quad \text{over } \widehat{\mathscr{B}} \in \mathscr{L} \quad \begin{bmatrix} \text{error}(w, \widehat{\mathscr{B}}) \\ \text{complexity}(\widehat{\mathscr{B}}) \end{bmatrix}$$

- tracing all optimal solutions requires hyper parameter
  1. λ — no physical meaning
  2. μ — bound on the error
  3. $(\mathtt{m}, \ell)$ — bound on the complexity