

# Lecture 6

## Least-squares applications

- least-squares data fitting
- growing sets of regressors
- system identification
- growing sets of measurements and recursive least-squares

6-1

### Least-squares data fitting

we are given:

- functions  $f_1, \dots, f_n : S \rightarrow \mathbf{R}$ , called *regressors* or *basis functions*
- *data or measurements*  $(s_i, g_i)$ ,  $i = 1, \dots, m$ , where  $s_i \in S$  and (usually)  $m \gg n$

**problem:** find coefficients  $x_1, \dots, x_n \in \mathbf{R}$  so that

$$x_1 f_1(s_i) + \dots + x_n f_n(s_i) \approx g_i, \quad i = 1, \dots, m$$

*i.e.*, find linear combination of functions that fits data

**least-squares fit:** choose  $x$  to minimize total square fitting error:

$$\sum_{i=1}^m (x_1 f_1(s_i) + \dots + x_n f_n(s_i) - g_i)^2$$

- using matrix notation, total square fitting error is  $\|Ax - g\|^2$ , where  $A_{ij} = f_j(s_i)$
- hence, least-squares fit is given by

$$x = (A^T A)^{-1} A^T g$$

(assuming  $A$  is skinny, full rank)

- corresponding function is

$$f_{\text{lsfit}}(s) = x_1 f_1(s) + \cdots + x_n f_n(s)$$

- applications:
  - interpolation, extrapolation, smoothing of data
  - developing simple, approximate model of data

## Least-squares polynomial fitting

**problem:** fit polynomial of degree  $< n$ ,

$$p(t) = a_0 + a_1 t + \cdots + a_{n-1} t^{n-1},$$

to data  $(t_i, y_i)$ ,  $i = 1, \dots, m$

- basis functions are  $f_j(t) = t^{j-1}$ ,  $j = 1, \dots, n$
- matrix  $A$  has form  $A_{ij} = t_i^{j-1}$

$$A = \begin{bmatrix} 1 & t_1 & t_1^2 & \cdots & t_1^{n-1} \\ 1 & t_2 & t_2^2 & \cdots & t_2^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & t_m & t_m^2 & \cdots & t_m^{n-1} \end{bmatrix}$$

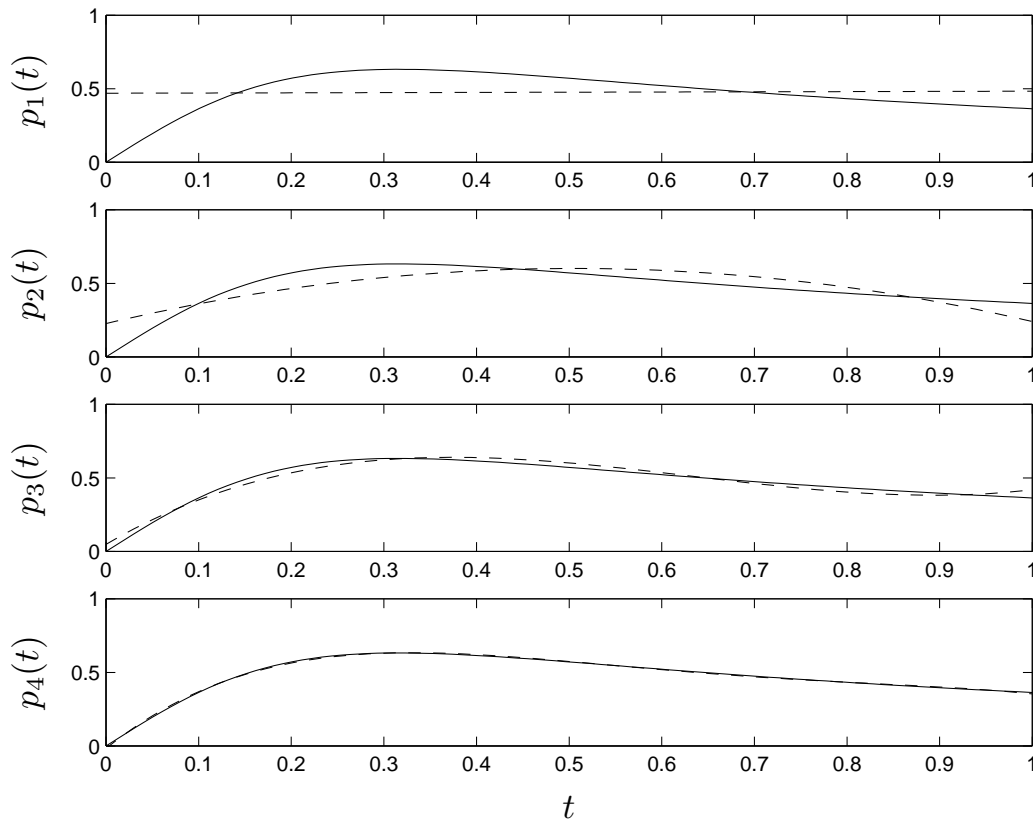
(called a *Vandermonde matrix*)

assuming  $t_k \neq t_l$  for  $k \neq l$  and  $m \geq n$ ,  $A$  is full rank:

- suppose  $Aa = 0$
- corresponding polynomial  $p(t) = a_0 + \cdots + a_{n-1}t^{n-1}$  vanishes at  $m$  points  $t_1, \dots, t_m$
- by fundamental theorem of algebra  $p$  can have no more than  $n - 1$  zeros, so  $p$  is identically zero, and  $a = 0$
- columns of  $A$  are independent, *i.e.*,  $A$  full rank

## Example

- fit  $g(t) = 4t/(1 + 10t^2)$  with polynomial
- $m = 100$  points between  $t = 0$  &  $t = 1$
- least-squares fit for degrees 1, 2, 3, 4 have RMS errors .135, .076, .025, .005, respectively



## Growing sets of regressors

consider *family* of least-squares problems

$$\text{minimize } \left\| \sum_{i=1}^p x_i a_i - y \right\|$$

for  $p = 1, \dots, n$

( $a_1, \dots, a_p$  are called *regressors*)

- approximate  $y$  by linear combination of  $a_1, \dots, a_p$
- project  $y$  onto  $\text{span}\{a_1, \dots, a_p\}$
- regress  $y$  on  $a_1, \dots, a_p$
- as  $p$  increases, get better fit, so optimal residual decreases

solution for each  $p \leq n$  is given by

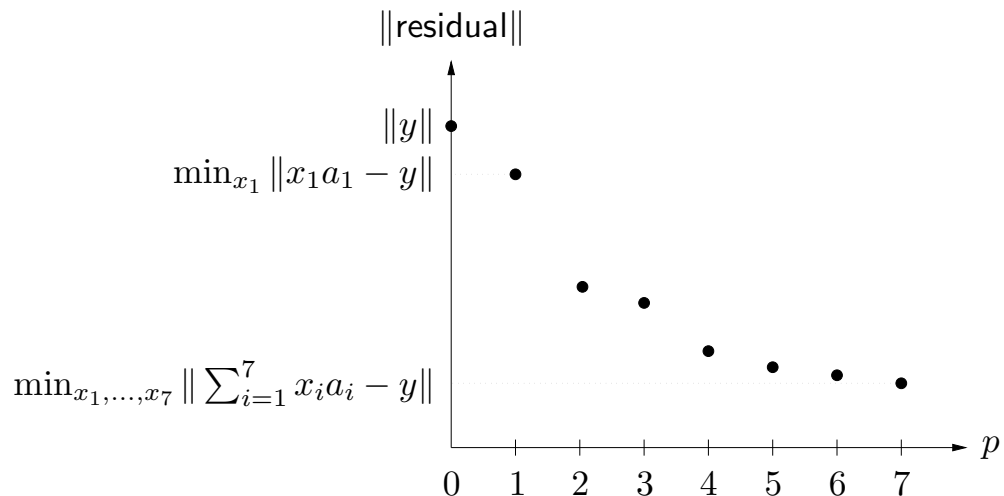
$$x_{\text{ls}}^{(p)} = (A_p^T A_p)^{-1} A_p^T y = R_p^{-1} Q_p^T y$$

where

- $A_p = [a_1 \cdots a_p] \in \mathbf{R}^{m \times p}$  is the first  $p$  columns of  $A$
- $A_p = Q_p R_p$  is the  $QR$  factorization of  $A_p$
- $R_p \in \mathbf{R}^{p \times p}$  is the leading  $p \times p$  submatrix of  $R$
- $Q_p = [q_1 \cdots q_p]$  is the first  $p$  columns of  $Q$

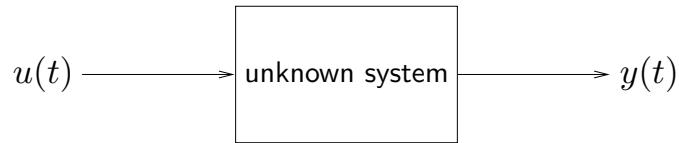
## Norm of optimal residual versus $p$

plot of optimal residual versus  $p$  shows how well  $y$  can be matched by linear combination of  $a_1, \dots, a_p$ , as function of  $p$



## Least-squares system identification

we measure input  $u(t)$  and output  $y(t)$  for  $t = 0, \dots, N$  of unknown system



**system identification problem:** find reasonable model for system based on measured I/O data  $u, y$

example with scalar  $u, y$  (vector  $u, y$  readily handled): fit I/O data with moving-average (MA) model with  $n$  delays

$$\hat{y}(t) = h_0 u(t) + h_1 u(t-1) + \dots + h_n u(t-n)$$

where  $h_0, \dots, h_n \in \mathbf{R}$

we can write model or predicted output as

$$\begin{bmatrix} \hat{y}(n) \\ \hat{y}(n+1) \\ \vdots \\ \hat{y}(N) \end{bmatrix} = \begin{bmatrix} u(n) & u(n-1) & \dots & u(0) \\ u(n+1) & u(n) & \dots & u(1) \\ \vdots & \vdots & \dots & \vdots \\ u(N) & u(N-1) & \dots & u(N-n) \end{bmatrix} \begin{bmatrix} h_0 \\ h_1 \\ \vdots \\ h_n \end{bmatrix}$$

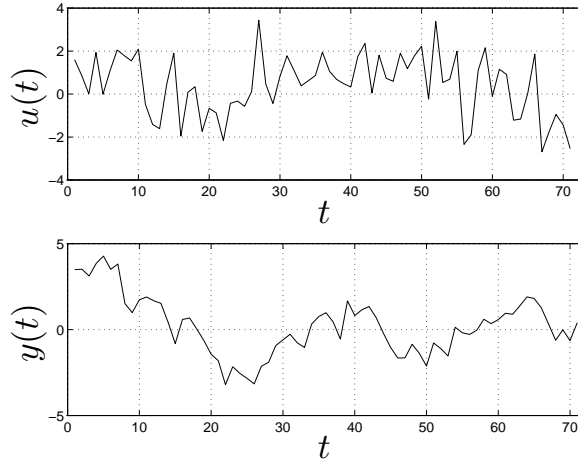
*model prediction error is*

$$e = (y(n) - \hat{y}(n), \dots, y(N) - \hat{y}(N))$$

**least-squares identification:** choose model (*i.e.*,  $h$ ) that minimizes norm of model prediction error  $\|e\|$

... a least-squares problem (with variables  $h$ )

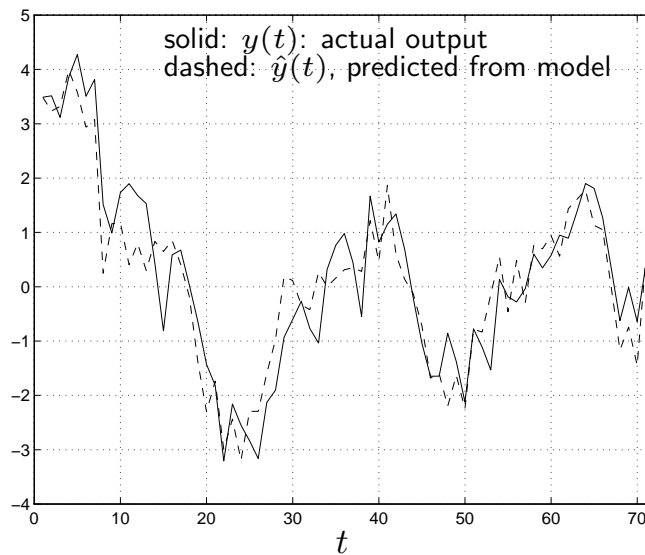
## Example



for  $n = 7$  we obtain MA model with

$$(h_0, \dots, h_7) = (.024, .282, .418, .354, .243, .487, .208, .441)$$

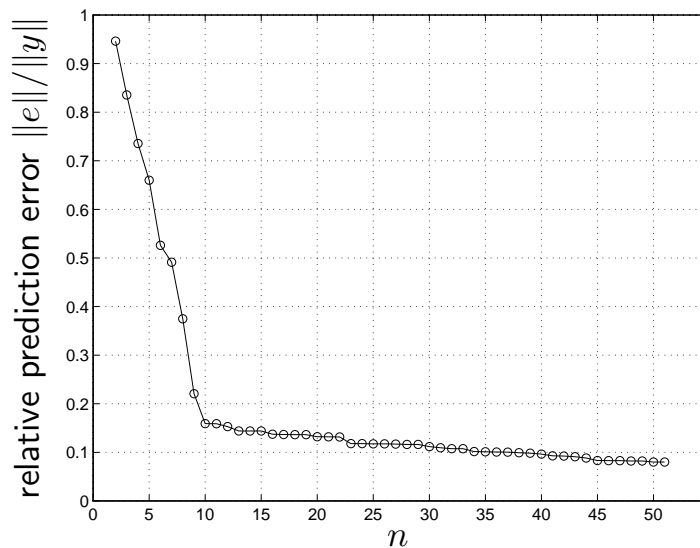
with relative prediction error  $\|e\|/\|y\| = 0.37$



# Model order selection

**question:** how large should  $n$  be?

- obviously the larger  $n$ , the smaller the prediction error *on the data used to form the model*
- suggests using largest possible model order for smallest prediction error

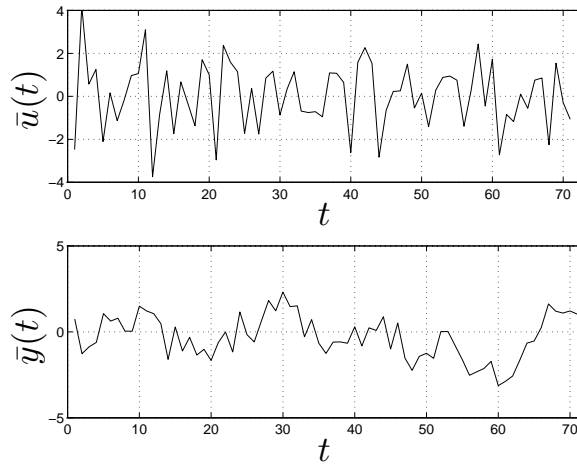


**difficulty:** for  $n$  too large the *predictive ability* of the model on *other I/O data* (from the same system) becomes worse

# Cross-validation

evaluate model predictive performance on another I/O data set *not used to develop model*

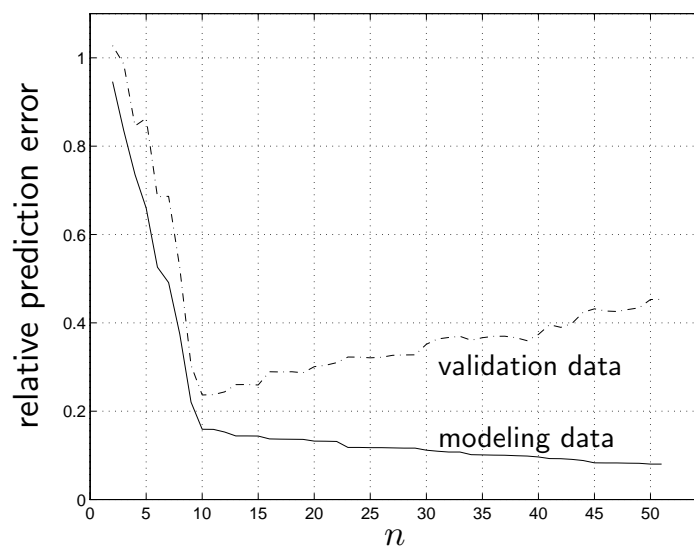
model validation data set:



Least-squares applications

6-17

now check prediction error of models (developed using *modeling data*) on *validation data*:

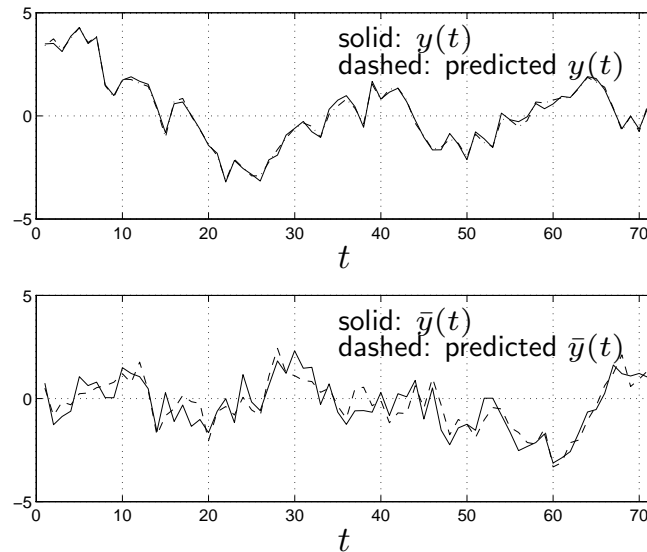


plot suggests  $n = 10$  is a good choice

Least-squares applications

6-18

for  $n = 50$  the actual and predicted outputs on system identification and model validation data are:



loss of predictive ability when  $n$  too large is called *model overfit* or *overmodeling*

## Growing sets of measurements

least-squares problem in 'row' form:

$$\text{minimize } \|Ax - y\|^2 = \sum_{i=1}^m (a_i^T x - y_i)^2$$

where  $a_i^T$  are the rows of  $A$  ( $a_i \in \mathbf{R}^n$ )

- $x \in \mathbf{R}^n$  is some vector to be estimated
- each pair  $a_i, y_i$  corresponds to one measurement
- solution is

$$x_{\text{ls}} = \left( \sum_{i=1}^m a_i a_i^T \right)^{-1} \sum_{i=1}^m y_i a_i$$

- suppose that  $a_i$  and  $y_i$  become available sequentially, *i.e.*,  $m$  increases with time

## Recursive least-squares

we can compute  $x_{\text{ls}}(m) = \left( \sum_{i=1}^m a_i a_i^T \right)^{-1} \sum_{i=1}^m y_i a_i$  recursively

- initialize  $P(0) = 0 \in \mathbf{R}^{n \times n}$ ,  $q(0) = 0 \in \mathbf{R}^n$
- for  $m = 0, 1, \dots$ ,

$$P(m+1) = P(m) + a_{m+1} a_{m+1}^T \quad q(m+1) = q(m) + y_{m+1} a_{m+1}$$

- if  $P(m)$  is invertible, we have  $x_{\text{ls}}(m) = P(m)^{-1} q(m)$
- $P(m)$  is invertible  $\iff a_1, \dots, a_m$  span  $\mathbf{R}^n$   
(so, once  $P(m)$  becomes invertible, it stays invertible)

## Fast update for recursive least-squares

we can calculate

$$P(m+1)^{-1} = (P(m) + a_{m+1} a_{m+1}^T)^{-1}$$

efficiently from  $P(m)^{-1}$  using the *rank one update formula*

$$(P + aa^T)^{-1} = P^{-1} - \frac{1}{1 + a^T P^{-1} a} (P^{-1} a)(P^{-1} a)^T$$

valid when  $P = P^T$ , and  $P$  and  $P + aa^T$  are both invertible

- gives an  $O(n^2)$  method for computing  $P(m+1)^{-1}$  from  $P(m)^{-1}$
- standard methods for computing  $P(m+1)^{-1}$  from  $P(m+1)$  is  $O(n^3)$

## Verification of rank one update formula

$$\begin{aligned} & (P + aa^T) \left( P^{-1} - \frac{1}{1 + a^T P^{-1} a} (P^{-1} a)(P^{-1} a)^T \right) \\ &= I + aa^T P^{-1} - \frac{1}{1 + a^T P^{-1} a} P (P^{-1} a)(P^{-1} a)^T \\ &\quad - \frac{1}{1 + a^T P^{-1} a} aa^T (P^{-1} a)(P^{-1} a)^T \\ &= I + aa^T P^{-1} - \frac{1}{1 + a^T P^{-1} a} aa^T P^{-1} - \frac{a^T P^{-1} a}{1 + a^T P^{-1} a} aa^T P^{-1} \\ &= I \end{aligned}$$