Lecture 12: Model Structure Determination via Stepwise Regression

Recall that the regression approach from the previous lecture postulated a model of the form

$$y = \theta_0 + \sum_{j=1}^{n} \theta_j \xi_j \tag{1}$$

where $\boldsymbol{\theta} = [\theta_0, \theta_1, \ldots, \theta_n]^{\mathrm{T}}$ is a vector of parameters to be determined and $\{\xi_1, \ldots, \xi_m\}$ are called regressors. A noise-corrupted version of the response variable is measured

$$z_k = \theta_0 + \sum_{j=1}^{n} \theta_j (\xi_j)_k + v_k \tag{2}$$

where v_k is assumed to be zero-mean Gaussian noise with variance σ_v^2. The matrix equivalents of the above expressions across all data points are

$$\boldsymbol{y} = \boldsymbol{\xi}\boldsymbol{\theta} \tag{3}$$

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} = \begin{bmatrix} 1 & (\xi_1)_1 & \cdots & (\xi_m)_1 \\ 1 & (\xi_1)_2 & \cdots & (\xi_m)_2 \\ \vdots & & & \\ 1 & (\xi_1)_N & \cdots & (\xi_m)_N \end{bmatrix} \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_m \end{bmatrix} \tag{4}$$

and

$$\boldsymbol{z} = \boldsymbol{\xi}\boldsymbol{\theta} + \boldsymbol{v} \tag{5}$$

$$\begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_N \end{bmatrix} = \begin{bmatrix} 1 & (\xi_1)_1 & \cdots & (\xi_m)_1 \\ 1 & (\xi_1)_2 & \cdots & (\xi_m)_2 \\ \vdots & & & \\ 1 & (\xi_1)_N & \cdots & (\xi_m)_N \end{bmatrix} \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_m \end{bmatrix} + \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_N \end{bmatrix} \tag{6}$$

The inherent assumption in this framework is that the *structure* of the model (i.e., the number and form of the model terms) is known. The problem was then to find the parameters within this class of model that best fit the data. However, in many cases, the model structure is also unknown and needs to be determined as part of system identification. In this lecture we will discuss the approach of *stepwise regression* which provides a systematic way to add/remove regressors from a pool of candidate regressors, while simultaneously optimizing their parameters, to arrive at a final model structure and set of model parameters. Naturally, during this process we must quantify how well a particular model fits the data and a mechanism to decide when to stop refining our model. We begin with some statistical background that is needed later on.

Statistical Modeling Metrics and Stopping Rules

Regression Sum of Squares. Let $\bar{z} = (1/N)\sum_{i=1}^{N} z_i$ denote the average value of the response variable measured over the set of N data points $\boldsymbol{z} = [z_1, \ldots, z_N]^{\mathrm{T}}$ and let $\hat{\boldsymbol{y}} = [\hat{y}_1, \ldots, \hat{y}_N]^{\mathrm{T}}$ denote the model-based estimate of the (noise-free) response variable. That is, if we have an estimate of the parameter $\hat{\boldsymbol{\theta}}$ then the predicted response variables are $\hat{y}_k = \hat{\theta}_0 + \sum_{j=1}^{n} \hat{\theta}_j (\xi_j)_k$. The following three sum-of-squares (SS) terms can be used to relate $\boldsymbol{z}$, $\bar{z}$ and $\hat{\boldsymbol{y}}$:

- Total sum of squares: quantifying variation between data and mean

$$SS_T = \sum_{i=1}^{N} [z_i - \bar{z}]^2 = \boldsymbol{z}^{\mathrm{T}}\boldsymbol{z} - N\bar{z}^2 \tag{7}$$

- Residual sum of squares: quantifying variation between data and model-predicted output

$$\begin{aligned} SS_E &= \sum_{i=1}^{N} [z_i - \hat{y}_i]^2 && (8)\\ &= (\boldsymbol{z} - \boldsymbol{\xi}\hat{\boldsymbol{\theta}})^{\mathrm{T}}(\boldsymbol{z} - \boldsymbol{\xi}\hat{\boldsymbol{\theta}}) && (9)\\ &= (\boldsymbol{z}^{\mathrm{T}} - \hat{\boldsymbol{\theta}}^{\mathrm{T}}\boldsymbol{\xi}^{\mathrm{T}})(\boldsymbol{z} - \boldsymbol{\xi}\hat{\boldsymbol{\theta}}) && (10)\\ &= \boldsymbol{z}^{\mathrm{T}}\boldsymbol{z} - \underbrace{\boldsymbol{z}^{\mathrm{T}}\boldsymbol{\xi}^{\mathrm{T}}\hat{\boldsymbol{\theta}}}_{\hat{\boldsymbol{\theta}}^{\mathrm{T}}\boldsymbol{\xi}\boldsymbol{z}} - \hat{\boldsymbol{\theta}}^{\mathrm{T}}\boldsymbol{\xi}^{\mathrm{T}}\boldsymbol{z} + \hat{\boldsymbol{\theta}}^{\mathrm{T}}\underbrace{\boldsymbol{\xi}^{\mathrm{T}}\boldsymbol{\xi}\hat{\boldsymbol{\theta}}}_{\boldsymbol{\xi}^{\mathrm{T}}\boldsymbol{z}} && (11)\\ &= \boldsymbol{z}^{\mathrm{T}}\boldsymbol{z} - 2\hat{\boldsymbol{\theta}}^{\mathrm{T}}\boldsymbol{\xi}\boldsymbol{z} + \hat{\boldsymbol{\theta}}\boldsymbol{\xi}^{\mathrm{T}}\boldsymbol{z} && (12)\\ &= \boldsymbol{z}^{\mathrm{T}}\boldsymbol{z} - \hat{\boldsymbol{\theta}}^{\mathrm{T}}\boldsymbol{\xi}^{\mathrm{T}}\boldsymbol{z} && (13) \end{aligned}$$

> **Aside:** Recall from our previous lecture on linear regression that after taking the partial derivative of the cost function J with respect to the parameter $\boldsymbol{\theta}$ and setting equal to zero (i.e., $\partial J/\partial\boldsymbol{\theta} = 0$) we had the final step which led to the pseudo-inverse
>
> $$\boldsymbol{\xi}^{\mathrm{T}}\boldsymbol{z} = \boldsymbol{\xi}^{\mathrm{T}}\boldsymbol{\xi}\hat{\boldsymbol{\theta}}$$
>
> This fact is used above in simplifying (11).

- Regression sum of squares: quantifying variation between model-predicted output and data-based mean

$$SS_R = \sum_{i=1}^{N} [\hat{y}_i - \bar{z}]^2 \tag{14}$$

which is related to the previous quantities by

$$\begin{aligned} SS_R &= SS_T - SS_E && (15)\\ &= \hat{\boldsymbol{\theta}}^{\mathrm{T}}\boldsymbol{\xi}^{\mathrm{T}}\boldsymbol{z} - N\bar{z}^2 && (16) \end{aligned}$$

Coefficient of determination. The coefficient of determination R^2 represents the proportion of the variation in the measured output that is explained by the model:

$$R^2 = \frac{SS_R}{SS_T} = \frac{\sum_{i=1}^{N}[\hat{y}_i - \bar{z}]^2}{\sum_{i=1}^{N}[z_i - \bar{z}]^2} \tag{17}$$

or

$$R^2 = 1 - \frac{SS_E}{SS_T} = 1 - \frac{\sum_{i=1}^{N}[z_i - \hat{y}_i]^2}{\sum_{i=1}^{N}[z_i - \bar{z}]^2} \tag{18}$$

If the predicted outputs match the data perfect, $\hat{y}_i = z_i$, then $R^2 = 1$. On the other hand if the predicted outputs are all equal to the mean of the data $\hat{y}_i = \bar{z}$ (a very naive model) then $R^2 = 0$. The R^2 value can also be negative for models that have worse predictions.

Pearson's Correlation Coefficient The above terms quantify overall fit with the model using all regressors for the predictions. Now we consider the correlation for a specific regressor. The correlation between a regressor ζ_j and a measured response variable z adjusted for the mean value is:

$$r_{jz} = \frac{S_{jz}}{\sqrt{S_{jj}S_{zz}}} = \frac{\text{(covariance of } j \text{ with } z)}{\sqrt{(\text{variance of } j)(\text{variance of } z)}} \tag{19}$$

where

$$S_{jz} = \sum_{i=1}^{N}[\zeta_j(i) - \bar{\zeta}_j][z(i) - \bar{z}]$$
$$S_{zz} = \sum_{i=1}^{N}[z(i) - \bar{z}]^2$$
$$\bar{\zeta}_j = \frac{1}{N}\sum_{i=1}^{N}\zeta_j(i) \qquad \text{(average of regressor)}$$

The correlation coefficients lie in the range $r_{jz} \in [-1, 1]$ and measure the degree to which regressor j is linearly correlated with the data z. A value of 1 indicates data vectors with identical normalized variations (i.e., the regressor is closely related to the response variable). A value of -1 indicates identical variations that differ only by a minus sign (again, closely related but with an inverse relationship). A value of zero indicates completely uncorrelated normalized variations.

Model Determination: Forward Selection

The basic idea of forward selection is to begin with a model that has no regressors, except a constant bias term

$$\hat{y} = \hat{\theta}_0 \, . \tag{20}$$

Using ordinary least-square from the previous lecture the best-fit θ_0 is estimated as

$$\hat{\theta}_0 = \boldsymbol{\xi}^{+}\boldsymbol{z} \tag{21}$$

where $\boldsymbol{z} = [z_1, \ldots, z_N]^{\mathrm{T}}$ is column vector of scalar data points and $\boldsymbol{\xi} = [1, \ldots, 1]^{\mathrm{T}}$ is a $N \times 1$ column vector of all ones and $^{+}$ is the pseudinverse. Next, we consider a pool of candidate regressors (determined by the user) and test all candidate regressors (one at a time) to determine which one to add. That is, for each candidate regressor ξ_j we compute the correlation coefficient r_{jz} with the dependent variable using (19). The regressor with the highest absolute value of r_{jz} is then considered for the model. The criterion for the selected regressor to be accepted into the model is that the partial F statistic must exceed a pre-determined value called F-to-enter, or F_{in}

$$F_0 = \frac{SS_R(\hat{\theta}_{p+j}) - SS_R(\hat{\theta}_p)}{\sigma_v^2} > F_{in} \tag{22}$$

where $SS_R(\hat{\theta}_p)$ is the regression sum of squares (16) using the p terms already in the model, $SS_R(\hat{\theta}_{p+j})$ is the regression sum of squared obtained by adding the jth regressor to the original p terms, and σ_v^2 is the measurement error variance. The purpose of (22) is to determine if the parameter being added, θ_j, leads to a reduction in the sum-of-square metric (16). Normalizing by the measurement variance insures that values of $F_0 >> 1$ indicate this change is not due to

random variations of the measurement. In general, the regressor ξ_j is added to the model with p terms if the *F statistic* is sufficiently large.

To evaluate the terms in (22) we must (temporarily) postulate a model of the form

$$\hat{y} = \hat{\theta}_0 + \hat{\theta}_1 \xi_1 \,, \tag{23}$$

(where we've abused notation and let ξ_1 denote the new regressor ξ_j) and optimize for $\boldsymbol{\theta} = [\theta_0, \theta_1]^{\mathrm{T}}$ using

$$\hat{\boldsymbol{\theta}} = \boldsymbol{\xi}^{+} \boldsymbol{z} \tag{24}$$

where

$$\boldsymbol{\xi} = \begin{bmatrix} 1 & (\xi_1)_1 \\ 1 & (\xi_1)_2 \\ \vdots & \\ 1 & (\xi_1)_N \end{bmatrix} . \tag{25}$$

The estimated parameter is used to generate the estimated output

$$\hat{\boldsymbol{y}} = \boldsymbol{\xi} \hat{\boldsymbol{\theta}} \tag{26}$$

that is required to evaluate (22).

After adding the first regressor it is removed from the pool of candidate regressors and the process repeats. However, rather than testing correlation (30) with $\boldsymbol{z}$ we subtract off the current model terms so that we are comparing candidate regressors with the *residual*. That is, we transform the data as follows:

$$\text{residual} = \text{data} - \text{model} \tag{27}$$

$$\boldsymbol{\epsilon} = \boldsymbol{z} - \hat{\boldsymbol{y}} \tag{28}$$

$$= \boldsymbol{z} - \boldsymbol{\xi} \boldsymbol{\theta} \tag{29}$$

and use

$$r_{j\epsilon} = \frac{S_{j\epsilon}}{\sqrt{S_{\epsilon\epsilon} S_{zz}}} = \frac{(\text{covariance of } j \text{ with } \epsilon)}{\sqrt{(\text{variance of } j)(\text{variance of } \epsilon)}} \tag{30}$$

Again, the remaining candidate regressors are used to evaluate a model of the form

$$\hat{y} = \theta_0 + \theta_1 \xi_1 + \theta_2 \xi_2 \,. \tag{31}$$

where ξ_2 is the candidate regressor. However, now the correlation coefficient $r_{j\epsilon}$ is used instead of r_{jz}. As before, the new regressor is accepted if (22) is satisfied. If no regressor meets the criteria (22) the process is terminated and the model is finalized.

Model Determination: Stepwise Regression

Stepwise regression improves upon forward selection by allowing the model to be re-assessed each time a new regressor is added. If a previously added regressor no longer contributes significantly it can be eliminated to reduce the size of the model. This may occur if a regressor added early on becomes redundant (e.g., because it is closely related) to another regressor added later. At the end of each forwards selection stage, each regressor is re-evalauted according to

$$F_0 = \frac{SS_R(\hat{\theta}_p) - SS_R(\hat{\theta}_{p-j})}{\sigma_v^2} < F_{out} \tag{32}$$

The values of F_{in} and F_{out} are chosen based on a desired confidence level and the number of data points. The values F_{in} and F_{out} can also be changed adaptively as more model parameters are added.

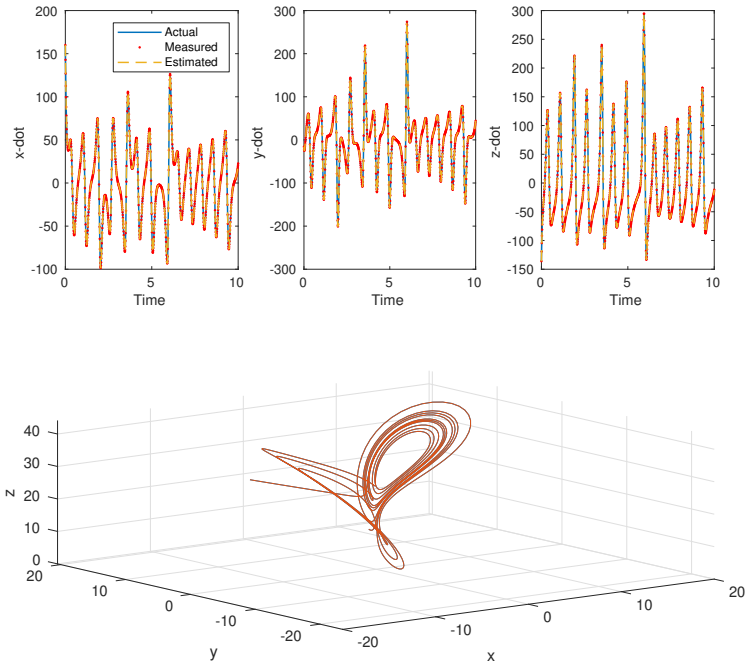
Example (Lorenz System). Consider the Lorenz system given by the nonlinear ODEs

$$\dot{x} = \sigma(y - x) \tag{33}$$

$$\dot{y} = x(\rho - z) - y \tag{34}$$

$$\dot{z} = xy - \beta z \tag{35}$$

This system exhibits chaotic behavior for certain parameter values of the constants σ, ρ, β. Suppose we obtain the derivative data $(\dot{x}, \dot{y}, \dot{z})$ from a simulation of the system lasting 10 seconds but are unaware of the system equations of motion (i.e., we have a block-box type of simulation). We can use stepwise regression to model the dynamics of each state derivative. That is, we can repeat the stepwise regression procedure three times to obtain models for $(\dot{x}, \dot{y}, \dot{z})$. In this example we simulate the above system for $\sigma = 10, \rho = 28$ and $\beta = 8/3$ from an initial condition of $[x, y, z]^{\mathrm{T}}(t_0) = [-8, 8, 27]^{\mathrm{T}}$ and use MATLAB `stepwiselm` tool. The explanatory variables are x, y, z and `stepwiselm` allows the user to form regressors that are products of these terms (i.e., we setup `stepwiselm` using the "quadratic" option). The actual, measured, and estimated values of the system are shown below.



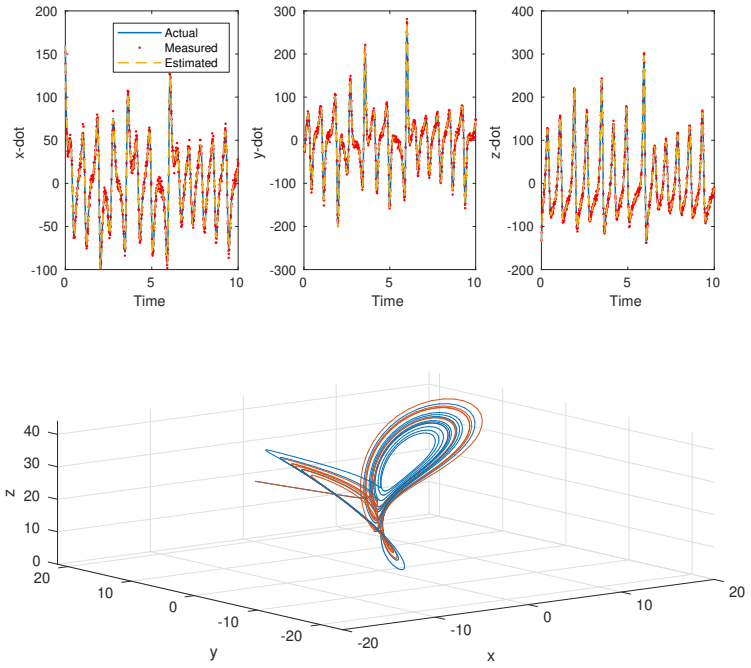
The estimated model using stepwise regression is

$$\dot{x} = -10x + 10y \tag{36}$$

$$\dot{y} = 28x - y - xz \tag{37}$$

$$\dot{z} = -2.6667z + xy \tag{38}$$

which exactly matches the true system. Note that this simulation was conducted without any noise and we ignored all the coefficients with magnitude less than 10^{-5}. If we add noise of $\sigma_v = 5$ to the derivative data we obtain the following result



with the model

$$\dot{x} = 0.101 - 9.958x + 9.939y \tag{39}$$

$$\dot{y} = 0.24 + 27.998x - 1.005y - 0.9996xz \tag{40}$$

$$\dot{z} = -0.12499 - 2.6551z + 1.0001xy \tag{41}$$

The code used in the above example is provided below:

```
clear; close all; clc;
Beta = [10; 28; 8/3]; % parameters
n = 3; % number of states
x0 = [−8; 8; 27]; % initial condition
tspan = [0.0:0.01:10]; % timesteps
sig1 = 5; % measurement noise
sig2 = sig1;
sig3= sig1;

% simulate nominal system (no noise)
options = [];
[t,X] = ode45(@(t,x) lorenz(t,x,Beta), tspan, x0, options);

% unpack  results into three explanatory variables
x1 = X(:,1);
x2 = X(:,2);
x3 = X(:,3);

% compute derivatives of
for i = 1:1:length(t)
```

```
xdot = lorenz(t,X(i,:)',Beta);
        z1(i) = xdot(1);
        z2(i) = xdot(2);
        z3(i) = xdot(3);
end
% add noise to derivative data (response variables)
z1n = z1 + randn(size(z1))*sig1;
z2n = z2 + randn(size(z2))*sig2;
z3n = z3 + randn(size(z3))*sig3;

% run the stepwise regression using
mdl1 = stepwiselm(X,z1n,'quadratic'); % x-dot model
mdl2 = stepwiselm(X,z2n,'quadratic'); % y-dot model
mdl3 = stepwiselm(X,z3n,'quadratic'); % z-dot model

% Note: stepwiselm produces table output with the final regressors/coefficients of
    the identified model. The coefficients can also be accessed via:
theta1 = mdl1.Coefficients.Estimate
theta2 = mdl2.Coefficients.Estimate
theta3 = mdl3.Coefficients.Estimate

%re-simulate the system using the identified  model
[tmdl,Xmdl] = ode45(@(t,x) lorenz_mdl(t,x,mdl1,mdl2,mdl3), tspan, x0, options);

% save outputs for comparison
x1mdl = Xmdl(:,1);
x2mdl = Xmdl(:,2);
x3mdl = Xmdl(:,3);

% the actual lorenz system
function dxdt = lorenz(t,xx,params)
x = xx(1);
y = xx(2);
z = xx(3);
sigma = params(1);
rho = params(2);
beta = params(3);
% the known dynamics
dxdt(1,1) = sigma*(y-x);
dxdt(2,1) = x*(rho-z)-y;
dxdt(3,1) = x*y - beta*z;
end

% the lorenz system obtained from stepwise regression. inputs include the three
    models for each state rate
```

```
function dxdt = lorenz_mdl(t,xx,mdl1,mdl2,mdl3)
x = xx(1);
y = xx(2);
z = xx(3);
% use the predict function to predict the state-rate of each variable
dxdt(1,1) = predict(mdl1,[x y z]);
dxdt(2,1) = predict(mdl2,[x y z]);
dxdt(3,1) = predict(mdl3,[x y z]);
end
```

References

- Morelli and Klein: Chapter 5.4
- Brunton and Kutz: Chapter 7.3 (SINDy)