Entropic Regression: Can we discover a dynamic that did not appear in our observations?

Illustrative example includes code usage.

This document and code is an illustrative example of sparse system identification with Entropic Regression (ER). For citation, and algorithm details, please refer to:

- Abd AlRahman R. AlMomani, Jie Sun, and Erik Bollt, "How Entropic Regression Beats the Outliers Problem in Nonlinear System Identification," Chaos 30, 013107 (2020).

Problem Setup

Here we concern about the problem of parameters estimation for the system:

$$\dot{x} = f(x, \beta)$$

where x is the state variable, is the vector field (underlying dynamic), and β is the set of parameters. If x is the only measured quantity, then $\dot{x}$ can be computed by the difference method or total variation method.

In this example, we will consider the Lorenz system as our subject system and the source of observed data.

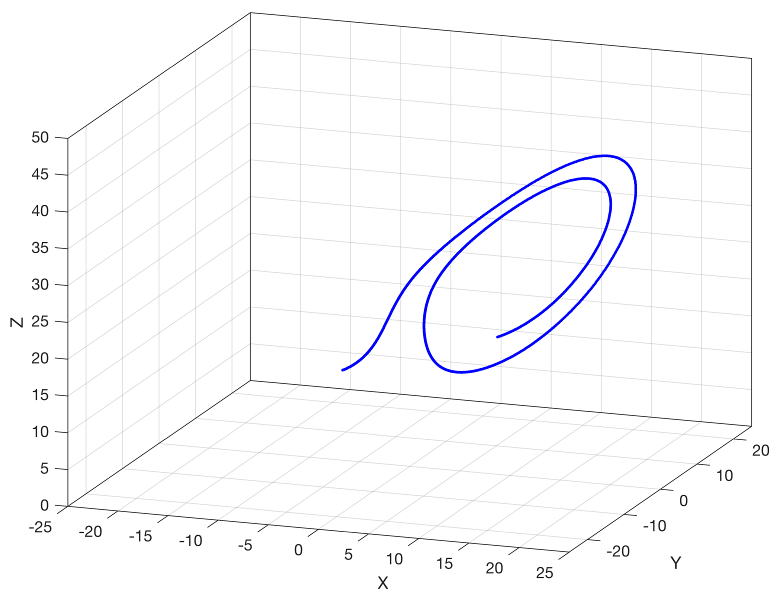
```
clearvars; close all; clc;
rng(5)
% Lorenz system ODE function
Lorenz = @(t,x) [10.*(x(2)-x(1));...
                 28.*x(1) - x(1).*x(3) - x(2);...
                 x(1).*x(2) - (8/3).*x(3);];
```

Now, we solve the forward system:

```
tao = 0.001;         % time step size
x0  = rand(3,1);     % initial condition
N   = 1500;          % number of observations to obtain

[~,X] = ode45(Lorenz, 0:tao:100, x0); %Transient
x0 = X(end,:)';
[tt,X] = ode45(Lorenz, 0:tao:(N+2)*tao, x0); %Sampled Data
```

Now, our data (state variable) stored in the matrix X, and shown in the following figure.

```
plot3(X(:,1),X(:,2),X(:,3),'.b')
view(20,20)
xlabel('X'); ylabel('Y'); zlabel('Z');
box on; grid on;
axis([-25 25 -25 25 0 50])
```




```

```

As you see, the data is sampled from a small part of one side of the Lorenz attractor, and the sampled data does not show the Lorenz system dynamics (the butterfly shape).

Noise and Outliers

In many approaches, the derivative computed from the clean and exact measurements in the matrix X, and the noise added only to the derivative (in analogy to the traditional linear algebra formulation $y = Ax + \epsilon$ where the noise appear only on y).

However, such approaches have poor reliability when we attempt to discover the underlying dynamics in the form of a set of ODEs, where the only measured quantity is the state variable X, and naturally, it should be the noisy measurements.

We assume a different kind of noise,

- A measurement noise, $\eta_1 \sim \mathcal{N}(0, \epsilon_1)$, that appear in all the sampled data.
- Outliers, $\eta_2 \sim \mathcal{N}(0, \epsilon_2)$, which is large magnitude (large deviation) noise that appear only on a fraction of the data. Each observation has a Bernoulli probability p to be corrupted with outliers.

In the following, we add the noise and outliers:

```
eps1 = 1e-4; % base noise std
eps2 = 0.1;  % outliers std
p    = 0.2;  % corruption probability

X = X + eps1*randn(size(X));

IX = rand(size(X,1),1)<= p;
% IX: Boolian string with the index of corrpted observations
X(IX,:) = X(IX,:) + eps2*randn(size(X(IX,:))); % add outliers noise
```

Now, X contain the noisy measurements with outliers.

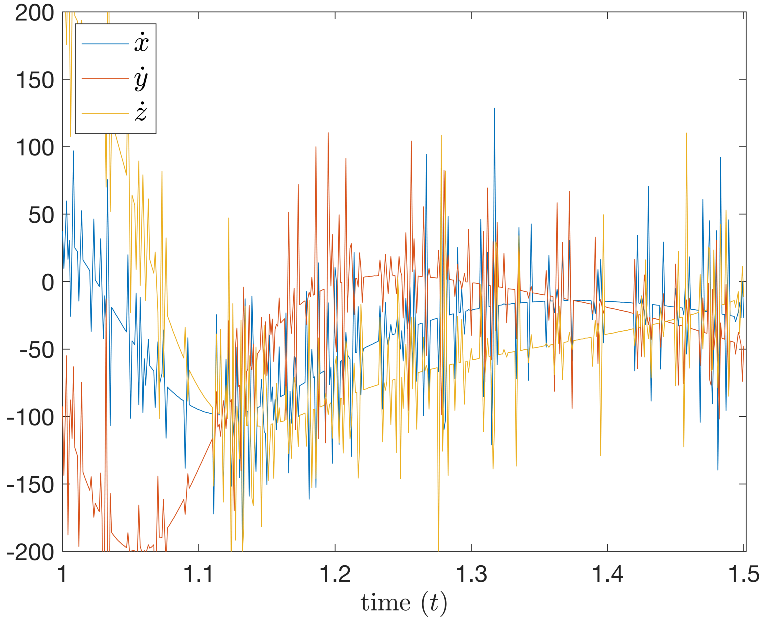
Derivative Estimation

To estimate the derivative, we will use the central difference method as the following:

```
Xdot = (1/(2*tao))*(X(3:end,:)-X(1:end-2,:));
X(1,:) = []; X(end,:) = [];
```

To see the effect of noise and outliers on the estimated derivative, we show the derivative time series as the following:

```
figure
plot(tt(1:end-2),Xdot,'-')
axis([1 max(tt) -200 200])
xlabel('time ($t$)','interpreter','latex');
set(gca,'FontSize',15)
legend('$\dot{x}$','$\dot{y}$','$\dot{z}$',...
       'interpreter','latex','Location','northwest',...
       'FontSize',18)
```



Inverse Problem

Now the problem is that:

Given X, and $\dot{X}$, find β.

We assume the power polynomial expansion of X as our candidate functions, and that expressed as:

$$\dot{X} = \sum_{i=1}^{K} \phi_i(X)\beta_i$$

and in a matrix form

$$\dot{X} = \Phi\beta$$

the power polynomial expansion of degree d to the matrix $X = [x, y, z]$ takes the form:

$$\Phi = [\mathbf{1, x, y, z, x^2, xy, xz, ..., z^d}]$$

We provide the function polyspace.m, which efficiently generate the basis matrix Φ from the state variable X.

```
Phi = polyspace(X,5); % power polynomial expansion of order 5
```

Entropic Regression

See the Matlab implementation and our published paper for the details of the ER algorithm. In the Matlab code attached, we provide an explanatory implementation that focuses on delivering the comments and explanations of the algorithm steps. However, a more efficient version with a wide range of options will be released soon with our ongoing paper on complex network coupling recovery.

The use of the Entropic Regression method is a straight forward call for the function erfit.m, with only the vector field and the basis matrix as inputs.

```
B = erfit(Phi, Xdot);
disp(B)
```

```
         0         0         0
   -9.9990   27.9339         0
   10.0014   -0.9725         0
         0         0   -2.6683
         0         0         0
         0         0    1.0002
         0   -0.9979         0
         0         0         0
         0         0         0
         0         0         0
         0         0         0
         0         0         0
         0         0         0
         0         0         0
         0         0         0
         0         0         0
         0         0         0
         0         0         0
         0         0         0
         0         0         0
         0         0         0
         0         0         0
         0         0         0
         0         0         0
         0         0         0
         0         0         0
         0         0         0
         0         0         0
         0         0         0
         0         0         0
         0         0         0
         0         0         0
         0         0         0
         0         0         0
         0         0         0
         0         0         0
         0         0         0
         0         0         0
         0         0         0
         0         0         0
         0         0         0
         0         0         0
         0         0         0
         0         0         0
         0         0         0
         0         0         0
         0         0         0
         0         0         0
         0         0         0
```

0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0

Prediction

The main objective of the mathematical model is to be able to reproduce the dynamic so that we can build predictions based on this model.

We showed that our sampled data (1500 observations) did not show the overall dynamic of the Lorenz system. In our paper, we showed how most state-of-the-art methods fail to reproduce the dynamic in such cases.

From the same initial condition, we solve the forward true model for the same length as of the sampled data:

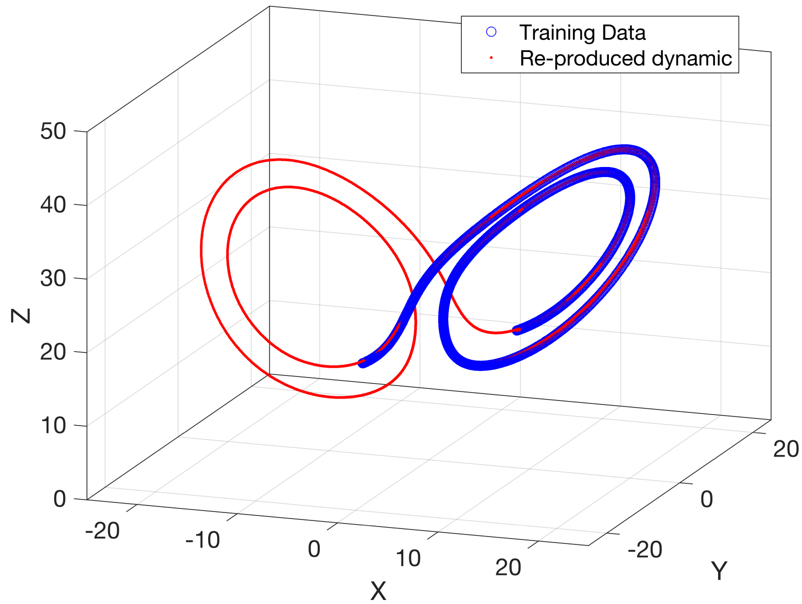
```
[~,Xtrue] = ode45(Lorenz, 0:tao:1.5, x0);
```

and, we will define a Parameters-Driven ODE function based on the results of the Entropic Regression, and solve the model fo 3 times longer time than the sampled data:

```
G = @(t,x) (polyspace(x',5)*B)';
[~,Xmodel] = ode45(G, 0:tao:4.5, x0); %Sampled Data
```

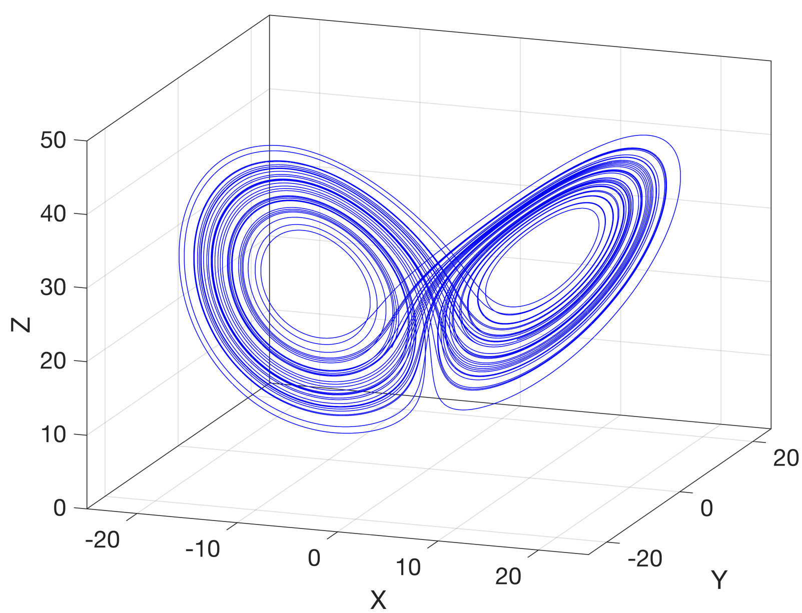
Now, we plot the true dynamics, and the reproduced dynamic:

```
figure

plot3(Xtrue(:,1),Xtrue(:,2),Xtrue(:,3),'ob')
hold on
plot3(Xmodel(:,1),Xmodel(:,2),Xmodel(:,3),'.r')

view(20,20)
xlabel('X'); ylabel('Y'); zlabel('Z');
box on; grid on;
axis([-25 25 -25 25 0 50])
legend('Training Data','Re-produced dynamic','Location','best')
set(gca,'FontSize',15)
```



And now, we show the long-time behavior of the model dynamics, which was able to reproduce the Lorenz system, without observing the overall behavior of the dynamic in the training data:

```
figure
[~,Xmodel] = ode45(G, 0:tao:50, x0);
plot3(Xmodel(:,1),Xmodel(:,2),Xmodel(:,3),'-b')
view(20,20)
xlabel('X'); ylabel('Y'); zlabel('Z');
box on; grid on;
axis([-25 25 -25 25 0 50])
set(gca,'FontSize',15)
```



Summary

In this supplementary material, we introduce illustrative example and explanatory code for the method of Entropic Regression, for sparse system identification based on causality inference. Complete and computationally efficient software (Matlab-Application) is under development and will be released soon.

Abd AlRahman AlMomani,

Research Associate (Postdoc),

Clarkson University, ECE,

Clarkson Center for Complex System Science (C^3S^2),

almomaniar@gmail.com, aaalmoma@clarkson.edu,

January 2020.