ReadMe-Automated_CEM_Computation_Library
This library is created to automatically compute the CEM and perform the permutation test on a set of input data for a set of potential functions selected by the user. This library includes the following files:

Library Contents Overview

calccem: Actual function to be used for automatic calculation of the CEM.

Usage: [rawCEM,adjustedCEM,Tmatrix]=calccem(xdata,time,potentialFunctions,perm)

Inputs:

xdata: matrix where each row is a state's trajectory with each column representing a given time instance.
time: the corresponding time vector for the system. Only required if time explicitly appears in the dynamics of the system being considered. Otherwise can be removed or input as empty.
potentialFunctions: anonymous function of the potential function vector **F** from [1].
perm: number of permutations to be considered for the permutation test.

Outputs:

rawCEM: Directly computed CEM as discussed in [1].
adjustedCEM: The CEM with terms termed insignificant by the permutation test or returned as negative values removed.
Tmatrix: The results of the significantly significant Causation Entropy value as determined by the permutation test

Notes: Makes use of Matlab's parallel computation toolbox. If user doesn't have it, simply change

driverfunction: Sample function to demonstrate usage of calccem. Calls function to generate dynamics and then compute corresponding CEM. Function is not needed if using calccem with independently generated data. See system description below for outline of system.

Usage: driverfunction

Inputs: none
Outputs: none

original_conditional_entropy: Function used to estimate conditional entropy (the necessary metric for causation entropy estimation). **Function is called by calccem and does not need to be used/modified by the user.**

Usage: see commented code for details:

Inputs: time series of appropriate structure as created by calccem
Outputs: conditional entropy used by calccem to estimate causation entropy

original_permutation_test: Function used to determine statistically relevant causation entropy value from the permutation test as discussed in [1]. **Function is called by calccem and does not need to be used/modified by the user.**

Usage: see commented code for details

Inputs:

percentage: Value of statistical certainty to be used for the permutation test between (0,1). 0.99 used for all works on this matter by the authors. Set in calccem.
perm: Number of permutations to be used for determining statistical significance. Authors typically use 75-100. The larger the number, the greater the consistency, but the increase comes with a significant increase in computational load.
Remainder of inputs are of structured data as computed by calccem. See commented code for details.

p_mkde: Function used to estimate underlying PDF. Based on code by Taesam Lee for Mutual Information toolbox available on mathworks website. Estimates based on methodology outlined in [2]. **Function is called by original_conditional_entropy and does not need to be used/modified by the user.** See commented code for details.

System Description

Generative Dynamics

The system considered was the following discrete time system (dynamics included in makedyn function in driverfunction):

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_{t+1} = \begin{bmatrix} 0.985 & 0.0125 & 0 \\ -0.0095 & 1.0035 & 0.01 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \sin(5t) \end{bmatrix}_t$$

Function Set Considered

The following function set is considered to demonstrate the ability to demonstrate the CEM's ability to identify unnecessary functions (in this case x_1^2).

$$\boldsymbol{F} = \begin{bmatrix} x_1 \\ x_2 \\ \sin(5t) \\ x_1^2 \end{bmatrix}_t$$

The corresponding CEM should thus have the following structure where 1 represents a nonzero causation entropy value and 0 a 0 causation entropy value.

$$\begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

Sources

[1] J. Elinger and J. Rogers, "Information Theoretic Causality Measures for System Identification of Mechanical Systems," *Journal of Computational and Nonlinear Dynamics,* vol. 13, no. 7, July 2018.

[2] Y.-I. Moon, B. Rajagopalan and U. Lall, "Estimation of mutual information using kernel density estimators," *Physical Review E,* vol. 52, no. 3, 1995.