Convergent Cross-Mapping and Causality Detection

McCracken,Weigel

January 29, 2014

Abstract

Convergent Cross-Mapping is a technique, introduced by Sugihara *et. al.* [Detecting Causality in Complex Ecosystems, 2012], reported to be "a necessary condition for causation" capable of distinguishing causality from correlation in sets of time series data [2012]. We will show that CCM correlations do not in general agree with intuitive concepts of "driving" and "response", and as such, relationships among CCM correlations should not be considered indicative of causality. It is shown that CCM correlations can, however, be used to identify asymmetrical prediction capability between pairs of time series data. We introduce a 2-vector called the "directed correlation" and present several examples of its use for identifying asymmetrical prediction capability. The sensitivity of CCM correlations (and consequently, directed correlations) on embedding dimensions and lag times will be discussed and mitigation will be presented.

1 Introduction

2 Convergent Cross-Mapping

Convergent cross-mapping (CCM) is introduced in [**?**, **?**] by Sugihara *et. al.* . CCM is described as a technique used to identify "causality" between time series and is intended to be useful in situations where Granger causality [] is known to be invalid (i.e. in dynamic systems that are "nonseperable"). The authors state that CCM is a "necessary condition for causation". **Expand this discussion more.** It is well known [] that Granger causality is unrelated to causality as it is typically understood in physics. It will be shown that a similar conclusion can be drawn regarding CCM causality.

CCM is closely related to simplex projection, introduced by Sugihara and May in [], which uses the points with the most similar histories to the point at t to predict the point at $t+1$. Similarly, CCM uses points with the most similar histories to a point $X(t)$ (in a time series X) to estimate the point $Y(t)$ (in a time series Y).

2.1 CCM Algorithm

It is elucidating tp partition the CCM algorithm into five distinct (though related) steps:

1. Create the shadow manifold for X, called M_X
2. Find the nearest neighbours to $X(t)$ in M_X
3. Use the nearest neighbours to create weights
4. Use the weights to estimate $Y(t)$, called $Y(t)|M_X$
5. Find the correlation between $Y(t)$ and $Y(t)|M_X$

The steps vary in complexity and are explained in more detail below.

2.1.1 Create M_X

Given an embedding dimension E, the shadow manifold of X, called M_X, is created by associating an E-dimensional vector to each point $X(t)$ that is constructed as $\vec{X}(t) = (X(t), X(t-\tau), X(t-2\tau), \ldots, X(t-(E-1)\tau)$. The first such vector is created at $t = 1+(E-1)\tau \equiv t_s$ and the last is at $t = L \equiv t_l$ where L is the time series length (or "library length"). The shadow manifold of X is the collection of all such vectors, i.e. $M_X = \{\vec{X}(t)|t \in [t_s, t_l]\}$. The time step τ is not discussed much in any of the Sugihara papers, and it appears to always be assumed that $\tau = 1$. We will follow that assumption throughout these notes unless specifically stated otherwise.

2.1.2 Find Nearest Neighbours

The minimum number of points required for a bounding simplex in an E-dimensional space is $E+1$ (find a non-Sugihara reference for this statement). Following this requirement, the $E+1$ nearest neighbours for a point $\vec{X}(t)$ on M_X (remember that this "point" on the shadow manifold is an E-dimensional vector of lagged time series points from X) are found. The distances d to these points and the times t at which they occur are recorded. Thus, the nearest neighbour search results in a set of distances $\{d_1, d_2, \ldots, d_{E+1}\}$ and an associated set of times $\{t_1, t_2, \ldots, t_{E+1}\}$ (where the subscript 1 denotes the closest neighbour, 2 denotes the next closest neighbour, etc.). The distances for a point $\vec{X}(t)$ are defined as

$$d_i = D\left(\vec{X}(t), \vec{X}(t_i)\right) \quad ,$$

where $D(\vec{a}, \vec{b})$ is the Euclidean distance between vectors $\vec{a}$ and $\vec{b}$ (implemented as `norm(a-b)` in the Matlab algorithm).

2.1.3 Create Weights

Each nearest neighbour will be used to find an associated weight. Define the unnormalized weights as

$$u_i = e^{-\frac{d_i}{d_1}} \quad .$$

In this way, each nearest neighbour will have a set of $E+1$ weights associated to the distance (and time) sets. The weights are defined as

$$w_i = \frac{u_i}{N} \quad ,$$

where the normalization factor is given as

$$N = \sum_j u_j \quad .$$

2.1.4 Find $Y(t)|M_X$

A point $Y(t)$ in Y can be estimated using the (normalized) distances to the points in X that have the most similar histories to the point $X(t)$ (i.e. using the weights calculated above). This estimate is calculated as

$$Y(t)|M_X = \sum_i w_i Y(t_i) \quad ,$$

where w_i are the weights calculated in the previous subsection and t_i are the times associated to the nearest neighbours (and, subsequently, the weights w_i).

2.1.5 Find the Correlation

Define the CCM correlation as

$$C_{YX} = \rho_{Y(t),Y(t)|M_X} \quad ,$$

where $\rho_{A,B}$ is the standard Pearson's correlation coefficient between A and B. It can be seen from the above algorithm that $X = Y \Rightarrow C_{YX} = C_{XY}$, but in general, $C_{YX} \neq C_{XY}$.

3 Embedding Dimension and Lag Time

3.1 Two Population Model

3.2 RL Circuit

4 Directed Correlation

5 Causality ...?

6 Conclusion