Problem 1: Can Catastrophic Events in Dynamical Systems be Predicted in Advance?

A related problem: Can future behaviors of time-varying dynamical systems be forecasted?

Early Warning?
Problem 2: Reverse Engineering of Complex Networks

Assumption: all nodes are externally accessible

Full network topology?
Problem 3: Detecting Hidden Nodes

No information is available from the black node. How can we ascertain its existence and its location in the network? How can we distinguish hidden node from local noise sources?
Basic Idea (1)

Dynamical system: \( \frac{dx}{dt} = F(x), \quad x \in \mathbb{R}^m \)

Goal: to determine \( F(x) \) from measured time series \( x(t) \)!

Power-series expansion of jth component of vector field \( F(x) \)

\[
[F(x)]_j = \sum_{l_1=0}^{n} \sum_{l_2=0}^{n} \ldots \sum_{l_m=0}^{n} (a_{j\ l_1l_2\ldots l_m}) x_1^{l_1} x_2^{l_2} \ldots x_m^{l_m}
\]

\( x_k - k\text{th component of } x \); Highest-order power: \( n \)

\( (a_{j\ l_1l_2\ldots l_m}) - \text{coefficients to be estimated from time series} \)

- \( (1+n)^m \) coefficients altogether

If \( F(x) \) contains only a few power-series terms, most of the coefficients will be zero.
Basic Idea (2)

Concrete example: \( m = 3 \) (phase-space dimension): \((x, y, z)\)

\( n = 3 \) (highest order in power-series expansion)

total \((1 + n)^m = (1 + 3)^3 = 64\) unknown coefficients

\[
[F(x)]_1 = (a_1)_{0,0,0}x^0y^0z^0 + (a_1)_{1,0,0}x^1y^0z^0 + \ldots + (a_1)_{3,3,3}x^3y^3z^3
\]

Coefficient vector \( a_1 = \begin{pmatrix} (a_1)_{0,0,0} \\ (a_1)_{1,0,0} \\ \ldots \\ (a_1)_{3,3,3} \end{pmatrix} - 64 \times 1 \)

Measurement vector \( g(t) = [x(t)^0y(t)^0z(t)^0, x(t)^1y(t)^0z(t)^0, \ldots, x(t)^3y(t)^3z(t)^3] \)

\( 1 \times 64 \)

So \( [F(x(t))]_1 = g(t) \cdot a_1 \)
Basic Idea (3)

Suppose $\mathbf{x}(t)$ is available at times $t_0, t_1, t_2, \ldots, t_{10}$ (11 vector data points)

\[ \frac{d\mathbf{x}}{dt}(t_1) = [\mathbf{F}(\mathbf{x}(t_1))]_1 = g(t_1) \cdot a_1 \]

\[ \frac{d\mathbf{x}}{dt}(t_2) = [\mathbf{F}(\mathbf{x}(t_2))]_1 = g(t_2) \cdot a_1 \]

\[ \ldots \]

\[ \frac{d\mathbf{x}}{dt}(t_{10}) = [\mathbf{F}(\mathbf{x}(t_{10}))]_1 = g(t_{10}) \cdot a_1 \]

Derivative vector $\mathbf{dX} = \begin{pmatrix} (dx/dt)(t_1) \\ (dx/dt)(t_2) \\ \vdots \\ (dx/dt)(t_{10}) \end{pmatrix}_{10 \times 1}$

Measurement matrix $\mathbf{G} = \begin{pmatrix} g(t_1) \\ g(t_2) \\ \vdots \\ g(t_{10}) \end{pmatrix}_{10 \times 64}$

We finally have $\mathbf{dX} = \mathbf{G} \cdot a_1$ or $\mathbf{dX}_{10 \times 1} = \mathbf{G}_{10 \times 64} \cdot (a_1)_{64 \times 1}$
Basic Idea (4)

\[ \begin{align*}
\text{d}X &= G \cdot a_1 \quad \text{or} \quad \text{d}X_{10 \times 1} = G_{10 \times 64} \cdot (a_1)_{64 \times 1} \\
\text{Reminder: } a_1 \text{ is the coefficient vector for the first dynamical variable } x.
\end{align*} \]

To obtain \([F(x)]_2\), we expand

\[ [F(x)]_2 = (a_2)_{0,0,0} x^0 y^0 z^0 + (a_2)_{1,0,0} x^1 y^0 z^0 + \ldots + (a_2)_{3,3,3} x^3 y^3 z^3 \]

with \(a_2\), the coefficient vector for the second dynamical variable \(y\). We have

\[ \text{d}Y = G \cdot a_2 \quad \text{or} \quad \text{d}Y_{10 \times 1} = G_{10 \times 64} \cdot (a_2)_{64 \times 1} \]

where

\[ \text{d}Y = \begin{pmatrix} 
(dy/dt)(t_1) \\
(dy/dt)(t_2) \\
\vdots \\
(dy/dt)(t_{10}) 
\end{pmatrix}_{10 \times 1} \]

Note: measurement matrix \(G\) is the same.

Similar expressions can be obtained for all components of the velocity field.
Compressive Sensing (1)

Look at
\[ dX = G \cdot a_1 \]

or
\[ dX_{10 \times 1} = G_{10 \times 64} \cdot (a_1)_{64 \times 1} \]

Note that \( a_1 \) is sparse - Compressive sensing!

Data/Image compression:
\( \Phi \): Random projection (not full rank)
x - sparse vector to be recovered

Goal of compressive sensing: Find a vector \( x \) with minimum number of entries subject to the constraint
\[ y = \Phi \cdot x \]
Compressive Sensing (2)

Find a vector $x$ with minimum number of entries subject to the constraint $y = \Phi \cdot x$: $l_1$-norm

Why $l_1$-norm? - Simple example in three dimensions

Predicting Catastrophe (1)

Henon map: \((x_{n+1}, y_{n+1}) = (1 - ax_n^2 + y_n, bx_n)\)

Say the system operates at parameter values: \(a = 1.2\) and \(b = 0.3\).

There is a chaotic attractor.

Can we assess if a catastrophic bifurcation (e.g., crisis) is imminent based on a limited set of measurements?

**Step 1: Predicting system equations**

Distribution of predicted values of ten power-series coefficients: constant, \(y, y^2, y^3, x, xy, xy^2, x^2, x^2y, x^3\)  

# of data points used: 8
Step 2: Performing numerical bifurcation analysis

Reconstructing Full Topology of Oscillator Networks (1)

A class of commonly studied oscillator-network models:

\[
\frac{dx_i}{dt} = F_i(x_i) + \sum_{j=1, j \neq i}^{N} C_{ij} \cdot (x_j - x_i) \quad (i = 1, \ldots, N)
\]

- dynamical equation of node i

N - size of network, \( x_i \in \mathbb{R}^m \), \( C_{ij} \) is the local coupling matrix

\[
C_{ij} = \begin{pmatrix}
C_{ij}^{1,1} & C_{ij}^{1,2} & \cdots & C_{ij}^{1,m} \\
C_{ij}^{2,1} & C_{ij}^{2,2} & \cdots & C_{ij}^{2,m} \\
\vdots & \vdots & \ddots & \vdots \\
C_{ij}^{m,1} & C_{ij}^{m,2} & \cdots & C_{ij}^{m,m}
\end{pmatrix}
\]

- determines full topology

If there is at least one nonzero element in \( C_{ij} \), nodes i and j are coupled.

Goal: to determine all \( F_i(x_i) \) and \( C_{ij} \) from time series.
Reconstructing Full Topology of Oscillator Networks (2)

\[
\mathbf{X} = \begin{pmatrix}
\mathbf{x}_1 \\
\mathbf{x}_2 \\
\vdots \\
\mathbf{x}_N
\end{pmatrix}_{N \times 1}
\]

- Network equation is \( \frac{d\mathbf{X}}{dt} = \mathbf{G}(\mathbf{X}) \), where

\[
[G(X)]_i = F_i(x_i) + \sum_{j=1, j \neq i}^{N} C_{ij} \cdot (x_j - x_i)
\]

- A very high-dimensional \((Nm\)-dimensional\) dynamical system;
- For complex networks (e.g., random, small-world, scale-free), node-to-node connections are typically sparse;
- In power-series expansion of \([G(X)]_i\), most coefficients will be zero - guaranteeing sparsity condition for compressive sensing.

Evolutionary-Game Dynamics

**Prisoner’s dilemma game**

<table>
<thead>
<tr>
<th></th>
<th>Cooperate</th>
<th>Defect</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cooperate</td>
<td>win-win</td>
<td>lose much-win much</td>
</tr>
<tr>
<td>Defect</td>
<td>win much-lose much</td>
<td>lose-lose</td>
</tr>
</tbody>
</table>

Strategies: cooperation \( S(C) = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \); defection \( S(D) = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \)

Payoff matrix: \( P(PD) = \begin{pmatrix} 1 & 0 \\ b & 0 \end{pmatrix} \) \( b \) - parameter

Payoff of agent \( x \) from playing PDG with agent \( y \): \( M_{x \leftarrow y} = S_x^T P S_y \)

For example,
- \( M_{C \leftarrow C} = 1 \)
- \( M_{D \leftarrow D} = 0 \)
- \( M_{C \leftarrow D} = 0 \)
- \( M_{D \leftarrow C} = b \)
Evolutionary Game on Network (Social and Economical Systems)

A network of agents playing games with one another:

Adjacency matrix \( A = \begin{pmatrix} \cdots & \cdots & \cdots \\ \cdots & a_{xy} & \cdots \\ \cdots & \cdots & \cdots \end{pmatrix} \) :
\[
\begin{cases} 
  a_{xy} = 1 & \text{if } x \text{ connects with } y \\
  a_{xy} = 0 & \text{if no connection}
\end{cases}
\]

Payoff of agent \( x \) from agent \( y \): \( M_{x\leftarrow y} = a_{xy} S_x^T P S_y \)

Time series of agents (Detectable)
(1) payoffs
(2) strategies
Compressive sensing
Full social network structure
Prediction as a CS Problem

Payoff of \( x \) at time \( t \): 
\[
M_x(t) = a_{x1}S_x^T(t)PS_1(t) + a_{x2}S_x^T(t)PS_2(t) + \cdots + a_{xN}S_x^T(t)PS_N(t)
\]

\[
Y = \begin{pmatrix}
M_x(t_1) \\
M_x(t_2) \\
\vdots \\
M_x(t_m)
\end{pmatrix} \quad X = \begin{pmatrix}
a_{x1} \\
a_{x2} \\
\vdots \\
a_{xN}
\end{pmatrix}
\]

\( X \): connection vector of agent \( x \) (to be predicted)

\[
\Phi = \begin{pmatrix}
S_x^T(t_1)PS_1(t_1) & S_x^T(t_1)PS_2(t_1) & \cdots & S_x^T(t_1)PS_N(t_1) \\
S_x^T(t_2)PS_1(t_2) & S_x^T(t_2)PS_2(t_2) & \cdots & S_x^T(t_2)PS_N(t_2) \\
\vdots & \vdots & \vdots & \vdots \\
S_x^T(t_m)PS_1(t_m) & S_x^T(t_m)PS_2(t_m) & \cdots & S_x^T(t_m)PS_N(t_m)
\end{pmatrix}
\]

\[
Y = \Phi \cdot X \quad Y, \Phi: \text{ from time series}
\]

Reverse Engineering of a Real Social Network

22 students play PDG together and write down their payoffs and strategies.

Friendship network

Observation: Large-degree nodes are not necessarily winners

Detecting Hidden Node

Idea

• Two green nodes: immediate neighbors of hidden node
• Information from green nodes is not complete
• Anomalies in the prediction of connections of green nodes

Locating Hidden Source in Complex Spreading Network Using Binary Time Series

Reconstruction of complex geospatial networks

(A) Time Series collection

\[ t_1 \quad t_2 \quad t_w \]

\[ X_1 \]

\[ X_2 \]

\[ \vdots \]

\[ X_N \]

(B) Compressive Sensing

\[ \begin{bmatrix} \tilde{x}_i(t_1) \\ \tilde{x}_i(t_2) \\ \vdots \\ \tilde{x}_i(t_w) \end{bmatrix} = \begin{bmatrix} A_i(t_1) & B_i(t_1) & C_i(t_1) \\ A_i(t_2) & B_i(t_2) & C_i(t_2) \\ \vdots & \vdots & \vdots \\ A_i(t_w) & B_i(t_w) & C_i(t_w) \end{bmatrix} \begin{bmatrix} \tilde{a}_i \\ \tilde{b}_i \\ \tilde{c}_i \end{bmatrix} \]

(C) Reconstructed Weight

(D) Reconstructed Delays

(E) Final geo-spatial network (2D)

(F) Final geo-spatial network (3D)

Discussion

1. Key requirement of compressive sensing – the vector to be determined must be sparse.

   Dynamical systems - three cases:
   - Vector field/map contains a few Fourier-series terms - Yes
   - Vector field/map contains a few power-series terms - Yes
   - Vector field /map contains many terms – not known

Ikeda Map: \[ F(x,y) = [A + B(x \cos \phi - y \sin \phi), B(x \sin \phi + y \cos \phi)] \]

   where \[ \phi = p - \frac{k}{1 + x^2 + y^2} \]

   - describes dynamics in an optical cavity

Mathematical question: given an arbitrary function, can one find a suitable base of expansion so that the function can be represented by a limited number of terms?
Discussion

2. Networked systems described by evolutionary games – Yes
3. Measurements of ALL dynamical variables are needed.

Outstanding issue
If this is not the case, say, if only one dynamical variable can be measured, the CS-based method would not work. Delay-coordinate embedding method? - gives only a topological equivalent of the underlying dynamical system (e.g., Takens’ embedding theorem guarantees only a one-to-one correspondence between the true system and the reconstructed system).

Data Analysis and Complex Systems

Data-based Research is becoming increasingly important for understanding, predicting, and controlling Complex Dynamical Systems