Compressive Sensing Based Prediction of Dynamical Systems and Complex Networks

Celso Grebogi
University of Aberdeen, UK
http://www.abdn.ac.uk/icsmb/people/profiles/grebogi

Collaborators:

Prof Ying-Cheng Lai, ASU & Univ. of Aberdeen
Dr Wenxu Wang, Beijing Normal University
Prof Younghae Do, Kyungpook University
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?
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?
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 \( F(x) \)

\[
[F(x)]_j = \sum_{l_1=0}^{n} \sum_{l_2=0}^{n} \ldots \sum_{l_m=0}^{n} (a_j)_{l_1l_2\ldots l_m} x_{l_1}^{l_1} x_{l_2}^{l_2} \ldots x_{l_m}^{l_m}
\]

\( x_k \) – kth component of \( x \); Highest-order power: \( n \)

\( (a_j)_{l_1l_2\ldots l_m} \) - 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.

W.-X. Wang, R. Yang, Y.-C. Lai, V. Kovanis, and C. Grebogi,
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 \( a_1 = \begin{pmatrix} (a_1)_{0,0,0} \\ (a_1)_{1,0,0} \\ \vdots \\ (a_1)_{3,3,3} \end{pmatrix} \) - 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 \( x(t) \) is available at times \( t_0, t_1, t_2, \ldots, t_{10} \) (11 vector data points)

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

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

\[\vdots\]

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

Derivative vector \( dX = \begin{pmatrix} (dx/dt)(t_1) \\ (dx/dt)(t_2) \\ \vdots \\ (dx/dt)(t_{10}) \end{pmatrix}_{10 \times 1} \); Measurement matrix \( G = \begin{pmatrix} g(t_1) \\ g(t_2) \\ \vdots \\ g(t_{10}) \end{pmatrix}_{10 \times 64} \)

We finally have \( dX = G \cdot a_1 \) or \( dX_{10 \times 1} = G_{10 \times 64} \cdot (a_1)_{64 \times 1} \)
Basic idea (4)

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

Reminder: \( a_1 \) is the coefficient vector for the first dynamical variable \( x \).

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

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

where

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

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

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

Look at
\[ dX = G \cdot a_1 \quad \text{or} \quad dX_{10 \times 1} = G_{10 \times 64} \cdot (a_1)_{64 \times 1} \]

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 $y = \Phi \cdot x$: $l_1 - \text{norm}$

Why $l_1 - \text{norm}$? - 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**

![Bar charts](chart)

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
Predicting catastrophe (2)

Step 2: Performing numerical bifurcation analysis

[Graph showing bifurcations with labeled axes and points: $y$ vs $a$, current operation point, and boundary crisis.]
Predicting catastrophe (3)

Examples of predicting continuous-time dynamical systems

Classical Lorenz system
\[
\begin{align*}
\frac{dx}{dt} &= 10y - 10x \\
\frac{dy}{dt} &= x(a - z) - y \\
\frac{dz}{dt} &= xy - \frac{2}{3}z
\end{align*}
\]

Classical Rossler system
\[
\begin{align*}
\frac{dx}{dt} &= -y - z \\
\frac{dy}{dt} &= x + 0.2y \\
\frac{dz}{dt} &= 0.2 + z(x - a)
\end{align*}
\]

# of data points used: 18
**Performance analysis**

![Standard map and Lorenz system graphs](image)

- $n_m$ – # of measurements
- $n_{nz}$ – # of non-zero coefficients; $n_z$ – # of zero coefficients
- ($n_{nz} + n_z$) – total # of coefficients to be determined
- $n_t$ – minimum # of measurements required for accurate prediction
Effect of noise

Henon map

![Graph](image)

\[ n_m = 8 \]
\[ (n_{n_z} + n_z) = 16 \]

Standard map

![Graph](image)

\[ n_m = 10 \]
\[ (n_{n_z} + n_z) = 20 \]

Predicting future attractors of time-varying dynamical systems (1)

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

\( p(t) \) - parameters varying slowly with time

\( T_M \) – measurement time period;

\( x(t) \) - available in time interval: \( t_M - T_M \leq t \leq t_M \)

Goal: to determine both \( F[x, p(t)] \) and \( p(t) \) from available time series \( x(t) \) so that the nature of the attractor for \( t > t_M \) can be assessed.

Power-series expansion

\[
[F(x)]_j = \sum_{l_1=0}^{n} \sum_{l_2=0}^{n} \ldots \sum_{l_m=0}^{n} (\alpha_j)_{l_1 \ldots l_m} x_1^{l_1} x_2^{l_2} \ldots x_m^{l_m} \sum_{w=0}^{v} (\beta_j)_w t^w
\]

\[
= \sum_{l_1, \ldots, l_m=0}^{n} \sum_{w=0}^{v} (c_j)_{l_1, \ldots, l_m; w} x_1^{l_1} x_2^{l_2} \ldots x_m^{l_m} \cdot t^w \Leftrightarrow \text{CS framework}
\]
Predicting future attractors of time-varying dynamical systems (2)

Formulated as a CS problem:

\[
\begin{align*}
\text{(a) Assumption} & \\
\begin{bmatrix}
    t^0: x \\
    t^1: x_t & y_t & \ldots & z^2_t
\end{bmatrix} & \\
\text{(b) Observation} & \\
\begin{bmatrix}
    x(t_1) & y(t_1) & \ldots & z(t_1)^2 \quad \vdots \\
    x(t_2) & y(t_2) & \ldots & z(t_2)^2 \\
    \vdots & \ddots & \ddots & \vdots \\
    x(t_M) & y(t_M) & \ldots & z(t_M)^2
\end{bmatrix} & \\
\begin{bmatrix}
    t_1 \\
    t_2 \\
    \vdots \\
    t_M
\end{bmatrix} & \\
\begin{bmatrix}
    x(t_1) t_1 & y(t_1) t_1 & \ldots & z(t_1)^2 t_1 \\
    x(t_2) t_2 & y(t_2) t_2 & \ldots & z(t_2)^2 t_2 \\
    \vdots & \ddots & \ddots & \vdots \\
    x(t_M) t_M & y(t_M) t_M & \ldots & z(t_M)^2 t_M
\end{bmatrix} & \\
\begin{bmatrix}
    c_1 \\
    c_2 \\
    \vdots \\
    c_N
\end{bmatrix} & \\
\begin{bmatrix}
    \dot{x}(t_1) \\
    \dot{x}(t_2) \\
    \vdots \\
    \dot{x}(t_M)
\end{bmatrix}
\end{align*}
\]
Predicting future attractors of time-varying dynamical systems (3)

Time-varying Lorenz system

\[
\begin{align*}
\frac{dx}{dt} &= -10(x - y) + k_1(t) \cdot y \\
\frac{dy}{dt} &= 28x - y - xz + k_2(t) \cdot z \\
\frac{dz}{dt} &= xy - (8/3)z + [k_3(t) + k_4(t)] \cdot y
\end{align*}
\]


k_1(t) = 0
k_2(t) = 0
k_3(t) = 0
k_4(t) = 0

k_1(t) = - t^2
k_2(t) = 0.5t
k_3(t) = t
k_4(t) = - 0.5t^2
Uncovering 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, \( x_i \in R^m \), \( C_{ij} \) 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.
Uncovering full topology of oscillator networks (2)

\[ X = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix}_{N \times 1} \quad \text{Network equation is} \quad \frac{dX}{dt} = G(X), \text{ where} \]

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

- 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

Example: 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</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 $y$: $M_{x\leftrightarrow y} = S_x^T P S_y$

For example, $M_{C\leftrightarrow C} = 1$
$M_{D\leftrightarrow D} = 0$
$M_{C\leftrightarrow D} = 0$
$M_{D\leftrightarrow C} = b$
Evolutionary game on network (social and economical networks)

A network of agents playing games with one another:

Adjacency matrix $A = \begin{pmatrix} \cdots & \cdots & \cdots \\ \cdots & a_{xy} & \cdots \\ \cdots & \cdots & \cdots \end{pmatrix}$:

- $a_{xy} = 1$ if $x$ connects with $y$
- $a_{xy} = 0$ if no connection

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 $t$: $M_x(t) = a_{x1}S_{x}^T(t)PS_1(t) + a_{x2}S_{x}^T(t)PS_2(t) + \cdots + a_{xN}S_{x}^T(t)PS_N(t)$

\[
Y = \begin{pmatrix}
M_x(t_1) \\
M_x(t_2) \\
\vdots \\
M_x(t_m)
\end{pmatrix}
\quad X = \begin{pmatrix}
a_{x1} \\
a_{x2} \\
\vdots \\
a_{xN}
\end{pmatrix}
\quad X : \text{connection vector of agent } x \quad \text{(to be predicted)}
\]

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

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

Success rate for model networks
Reverse engineering of a real social network

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

Friendship network

Experimental record of two players

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

Distinguishing between effects of hidden node and noise (1)
Distinguishing between effects of hidden node and noise (2)

Cancellation ratio

Variance in predicted coefficient vector

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 \( \phi \equiv p - \frac{k}{1 + x^2 + y^2} \) - 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?
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).
4. In Conclusion, much work is needed!