Analysis of Network Time Series

Matt Nunes

Lancaster University

with
Marina Knight and Guy Nason

Newton Institute, January 2014
Many multivariate time series we encounter in practice will be observed on a network / graph, e.g.

- transport data
- computer traffic
- social media
Many multivariate time series we encounter in practice will be observed on a network / graph, e.g.

- transport data
- computer traffic
- social media

Data summary:

- time series observed on (large) network
- network is real, or can be constructed
- aim is efficient analysis e.g. **forecasting the series**
Toy example: Cases of Mumps 2005

Weekly cases of Mumps at county level for 2005 ($T = 1, \ldots, 52$. Number of Counties: $P = 47$).

data source: Health Protection Agency (thanks to D. Harding & D. DeAngelis).
What is the data like?
Hourly wind speeds collected at UK Met. Office weather stations ($T = 721$ records, $P = 102$ stations).

Data source: British Atmospheric Data Centre (BADC) / Centre for Environmental Data Archival (CEDA).
Correlation: Scampton and Rochdale

ACF

Lag

381

0 5 10 15 20 25
0.0 0.2 0.4 0.6 0.8 1.0

381 & 1125

0 5 10 15 20 25
0.0 0.2 0.4 0.6 0.8 1.0

1125 & 381

−25 −20 −15 −10 −5 0
0.0 0.2 0.4 0.6 0.8 1.0

1125

0 5 10 15 20 25
0.0 0.2 0.4 0.6 0.8 1.0

Matt Nunes
Analysis of Network Time Series
Mumps / wind data does not come with a “natural” network.
Where is the Network?

Mumps / wind data does not come with a “natural” network.

- Can use a tree structure to link weather stations by *geography*, or county borders to link data.
Where is the Network?

Mumps / wind data does not come with a “natural” network.

- Can use a tree structure to link weather stations by geography, or county borders to link data.

Other possible ways to construct networks:
- link with transport corridors, e.g. rail connections, roads etc
- “friends” / followers in social media
- academic collaborations (common papers, number of citations)
Network structure using distances between county towns.
Weather stations network created by minimal spanning tree.
Analysis question

Characteristics of network data:
- complex structure over time
- lots of dependence between series
- possible redundancy in the system
Analysis question

Characteristics of network data:
- complex structure over time
- lots of dependence between series
- possible redundancy in the system

- Can we capture the network structure?
- Can we reduce the dimension of the analysis problem?

Trick:
transform complex data into simpler time series for data analysis (dimension reduction) with lifting schemes.
Characteristics of network data:
- complex structure over time
- lots of dependence between series
- possible redundancy in the system

Can we capture the network structure?

Can we reduce the dimension of the analysis problem?

Trick:
transform complex data into \textit{simpler} time series for data analysis (dimension reduction) with \textbf{lifting schemes}
Lifting schemes (Sweldens 1995) can provide efficient representations for data arising on complex data domains e.g. graphs / networks (“wavelet transform on a network”).
Dimension reduction with lifting

Lifting schemes (Sweldens 1995) can provide efficient representations for data arising on complex data domains e.g. graphs / networks (“wavelet transform on a network”).

Why lifting?

- Wavelets are well-known for their ability to **decorrelate** systems (for decorrelation properties of lifting for time series, see KNN2014 in preparation)

- Lifting schemes can **naturally account for network structure**

- They are **computationally fast and invertible**

- \textbf{Split}: Pick network node to be ‘lifted’.
Network lifting algorithm: \( \text{NetTree}^1 \)

- **Predict**: Use function values at neighbouring nodes \( \mathcal{J}_i \) to predict \( c_i \) using distance-weighted average. The **lifted** coefficient is the residual

\[
d_i = c_i - \hat{c}_i = c_i - \sum_{j \in \mathcal{J}_i} a_{i,j} c_j.
\]

---

**Network lifting algorithm:** *NetTree*¹

- **Update:** Remove node $i$ and redistribute lost signal content to neighbours using the lifted coefficient $d_i$.

---

Network lifting algorithm: NetTree\(^1\)

- Repeat 1–3 until no network nodes are left.

  We get

  \[ Y = \mathcal{W}X. \]

---

Proposed dimension reduction procedure:

For each timepoint \( t = 1, \ldots, T \), perform \textit{NetTree} lifting algorithm

Get new time series in the wavelet domain:
\[ Y_t = W(X_t) \]

wavelet coefficients series: \( P_{-Nc} \)
scale coefficients series: \( N_{c} \)

The transform takes into account network structure.

The resulting time series object is sparse.

What does this sparsity give us?
Proposed dimension reduction procedure:
For each timepoint \( t = 1, \ldots, T \), perform NetTree lifting algorithm

- Get new time series in the wavelet domain:

\[
Y_t = \mathcal{W}(X_t),
\]

wavelet coefficients series: \( P - N_c \) series;
scaling coefficients series: \( N_c \)
Dimension reduction for network time series

Proposed dimension reduction procedure:

For each timepoint \( t = 1, \ldots, T \), perform NetTree lifting algorithm

- Get new time series in the wavelet domain:
  \[
  Y_t = \mathcal{W}(X_t),
  \]
  wavelet coefficients series: \( P - N_c \) series;
  scaling coefficients series: \( N_c \)

- the transform takes into account network structure
- the resulting time series object is \textit{sparse}
Proposed dimension reduction procedure:

For each timepoint \( t = 1, \ldots, T \), perform NetTree lifting algorithm

- Get new time series in the wavelet domain:
  \[ Y_t = \mathcal{W}(X_t), \]
  - wavelet coefficients series: \( P - N_c \) series;
  - scaling coefficients series: \( N_c \)

- the transform takes into account network structure
- the resulting time series object is \textit{sparse}

What does this sparsity give us?
Mumps: Correlation before lifting

**Durham**

**Durham & Tyne and Wear**

**Tyne and Wear & Durham**

**Tyne and Wear**

Acyclic Correlation Function (ACF) plots for different regions showing lagged correlations.
Mumps: Correlation after lifting

Durham

Durham & Tyne and Wear

Tyne and Wear & Durham

Tyne and Wear

ACF

Lag

0 2 4 6 8 10 12 14

−0.2 0.2 0.6 1.0

Lag

0 2 4 6 8 10 12 14

−0.2 0.2 0.6 1.0

Lag

−14 −10 −8 −6 −4 −2 0

−0.2 0.2 0.6 1.0

Lag

0 2 4 6 8 10 12 14

−0.2 0.2 0.6 1.0

Lag

−0.2 0.2 0.6 1.0

Lag

−14 −10 −8 −6 −4 −2 0

−0.2 0.2 0.6 1.0

Lag

0 2 4 6 8 10 12 14

−0.2 0.2 0.6 1.0

Lag

0 2 4 6 8 10 12 14

−0.2 0.2 0.6 1.0

Lag

0 2 4 6 8 10 12 14

−0.2 0.2 0.6 1.0

Lag

0 2 4 6 8 10 12 14

−0.2 0.2 0.6 1.0

Lag

−0.2 0.2 0.6 1.0

Lag

−14 −10 −8 −6 −4 −2 0

−0.2 0.2 0.6 1.0

Lag

0 2 4 6 8 10 12 14

−0.2 0.2 0.6 1.0

Lag

0 2 4 6 8 10 12 14

−0.2 0.2 0.6 1.0

Lag

0 2 4 6 8 10 12 14

−0.2 0.2 0.6 1.0

Lag

0 2 4 6 8 10 12 14

−0.2 0.2 0.6 1.0

Lag

0 2 4 6 8 10 12 14

−0.2 0.2 0.6 1.0

Lag

0 2 4 6 8 10 12 14

−0.2 0.2 0.6 1.0

Lag

0 2 4 6 8 10 12 14

−0.2 0.2 0.6 1.0

Lag

0 2 4 6 8 10 12 14

−0.2 0.2 0.6 1.0

Lag

0 2 4 6 8 10 12 14

−0.2 0.2 0.6 1.0

Lag

0 2 4 6 8 10 12 14

−0.2 0.2 0.6 1.0

Lag
Shobdon Airf. and Coleshill: ACF (after lifting)

![ACF plots for 669, 669 & 19187, 19187 & 669, and 19187]

Matt Nunes
Analysis of Network Time Series
A Model

Massive decorrelation across network and in time...

Let $G$ be the graph (network), and let $S(G)$ be set of scaling function coefficient nodes. Then divide the wavelet coefficients into two sets:

- $Z(G)$ set of nodes whose series are white noise.
- $Q(G)$ set of nodes whose series are not white noise.

Can model $Q(G)$ as appropriate, e.g. low-order ARMA.
Massive decorrelation across network and in time...

Let $\mathcal{G}$ be the graph (network), and let $S(\mathcal{G})$ be set of scaling function coefficient nodes.
Massive decorrelation across network and in time...

Let $G$ be the graph (network), and let $S(G)$ be set of scaling function coefficient nodes.

Then divide the wavelet coefficients into TWO sets:

- $Z(G)$ set of nodes whose series are white noise.
- $Q(G)$ set of nodes whose series are not white noise.
Massive decorrelation across network and in time...

Let $\mathcal{G}$ be the graph (network), and let $S(\mathcal{G})$ be set of scaling function coefficient nodes.

Then divide the wavelet coefficients into TWO sets:

- $\mathcal{Z}(\mathcal{G})$ set of nodes whose series are white noise.
- $\mathcal{Q}(\mathcal{G})$ set of nodes whose series are not white noise.
Massive decorrelation across network and in time...

Let $\mathcal{G}$ be the graph (network), and let $S(\mathcal{G})$ be set of scaling function coefficient nodes.

Then divide the wavelet coefficients into TWO sets:

- $Z(\mathcal{G})$ set of nodes whose series are white noise.
- $Q(\mathcal{G})$ set of nodes whose series are not white noise.

Can model $Q(\mathcal{G})$ as appropriate, e.g. low-order ARMA.
A Model

For the Mumps Data:

Scaling coefficients:
\[ S(G) = \{30, 47\} = \{\text{"NorthYorkshire"}, \text{"Wiltshire"}\}. \]
For the Mumps Data:

Scaling coefficients:
\[
S(\mathcal{G}) = \{30, 47\} = \{\text{"NorthYorkshire"}, \text{"Wiltshire"}\}.
\]

Of the remaining 45 time series:

28 are deemed to be white noise \(Z(\mathcal{G})\);
17 are low order ARMA, \(Q(\mathcal{G})\).

That is over 60% of the series are white noise.
A Model

For the Mumps Data:

Scaling coefficients:
\[ S(\mathcal{G}) = \{30, 47\} = \{\text{"NorthYorkshire"}, \text{"Wiltshire"}\}. \]

Of the remaining 45 time series:

- 28 are deemed to be white noise \( Z(\mathcal{G}) \);
- 17 are low order ARMA, \( Q(\mathcal{G}) \).
A Model

For the Mumps Data:

Scaling coefficients:
\[ S(G) = \{30, 47\} = \{"NorthYorkshire", "Wiltshire"\}. \]

Of the remaining 45 time series:

- 28 are deemed to be white noise \( Z(G) \);
- 17 are low order ARMA, \( Q(G) \).

That is over 60% of the series are white noise.
Potential benefit for forecasting

Decorrelation across network means can treat series **separately.**
Potential benefit for forecasting

Decorrelation *across* network means can treat series *separately*.

Forecasting procedure:

1. Perform network lifting dimension reduction procedure.
Potential benefit for forecasting

Decorrelation *across* network means can treat series *separately*.

Forecasting procedure:

1. Perform network lifting dimension reduction procedure.
2. (Few) scaling coefficient series in $S(\mathcal{G})$ can be extrapolated forward.
   - White noise processes in $\mathcal{Z}(\mathcal{G})$: extremely easy to predict (i.e. the mean).
   - $Q(\mathcal{G})$: use favourite forecasting method, e.g. Box-Jenkins for ARMA processes.
Potential benefit for forecasting

Decorrelation across network means can treat series separately.

Forecasting procedure:

1. Perform network lifting dimension reduction procedure.
2. (Few) scaling coefficient series in $S(G)$ can be extrapolated forward.
   - White noise processes in $Z(G)$: extremely easy to predict (i.e. the mean).
   - $Q(G)$: use favourite forecasting method, e.g. Box-Jenkins for ARMA processes.
3. Then invert the lifting transform to obtain forecasts for the original series.
Improved Forecasting Performance

- Forecasting performance improves when compared to time domain forecasting (ignoring network structure) $\sim 20\%$.
- Additional improvement gained by bootstrapping over lifting orderings (“nondecimation”) and aggregating forecasts.
Developed a new technique for network time series data reduction, which takes into account network structure.

- decorrelation → much simpler modelling and forecasting in the transform domain.

- Preliminary results encouraging.
Summary

- Developed a new technique for network time series data reduction, which takes into account network structure
- Decorrelation $\rightarrow$ much simpler modelling and forecasting in the transform domain
- Preliminary results encouraging.

Open questions:
- Choice of network
- Optimal / adaptive lifting strategies for maximal decorrelation effect
- Other analysis tasks, e.g. changepoints?
