Empirical Dynamic Modelling

Automatic Causal Inference and Forecasting

Dr Patrick Laub

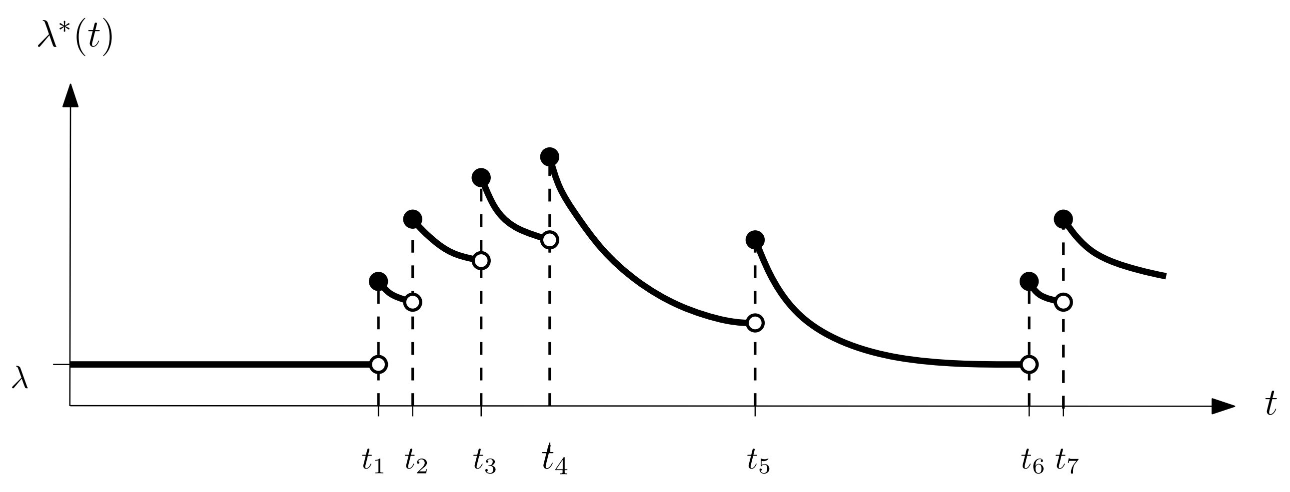
Time-Series and Forecasting Symposium

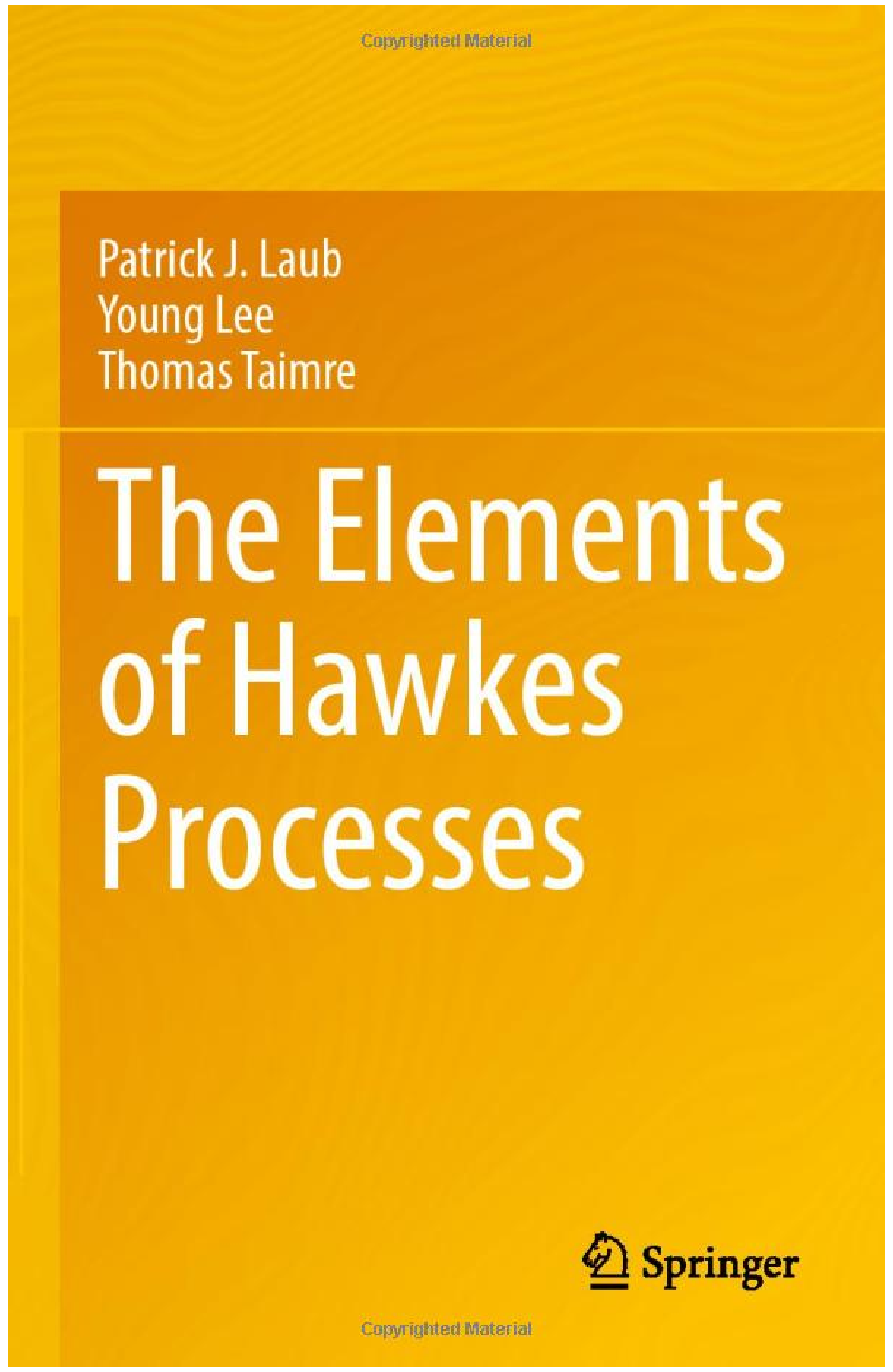
December 2, 2022

Introduction

About me

- Software engineer & maths (UQ)
- PhD in computational applied probability (Aarhus)
- Actuarial science post-doc (Lyon)
- Research software engineer (Uni Melbourne)
- Actuarial science lecturer (UNSW)





Other unrelated past work

approxbayescomp Python Package

Approximate Bayesian Computation Python Package

Package Description

Approximate Bayesian Computation (ABC) is a statistical method to fit a Bayesian model to data when the likelihood function is hard to compute. The `approxbayescomp` package implements an efficient form of ABC — the *sequential Monte Carlo (SMC)* algorithm. While it can handle any general statistical problem, we built in some models so that fitting insurance loss distributions is particularly easy.

Installation

To install simply run

```
pip install -U approxbayescomp
```

Soon, it will be possible to install using `conda`; at that point the preferred

Deep Learning for Actuaries

Lecture slides from UNSW's ACTL3143 & ACTL5111 courses

AUTHOR
Dr Patrick Laub

Overview

These are the lecture slides from my recent "Deep Learning for Actuaries" courses (coded ACTL3143 & ACTL5111) at UNSW. They can be used to see what topics I covered in these courses. The slides are not intended to be used to learn deep learning from scratch. For that, you need to attend the lectures & complete the assessment.

Lecture slides

- Course overview slides, source

Goal: automatic causal inference

```r
df <- read.csv("chicago.csv")
head(df)
#>   Time Temperature Crime
#> 1    1       24.08  1605
#> 2    2       19.04  1119
#> 3    3       28.04  1127
#> 4    4       30.02  1154
#> 5    5       35.96  1251
#> 6    6       33.08  1276

library(fastEDM)

crimeCCMCausesTemp <- easy_edm("Crime", "Temperature", data=df)
#> ✖ No evidence of CCM causation from Crime to Temperature found.

tempCCMCausesCrime <- easy_edm("Temperature", "Crime", data=df)
#> ✔ Some evidence of CCM causation from Temperature to Crime found.
```

The Stata Journal (2021)
21, Number 1, pp. 220–258
DOI: 10.1177/1536867X211000030

Beyond linearity, stability, and equilibrium: The edm package for empirical dynamic modeling and convergent cross-mapping in Stata

Jinjing Li
University of Canberra

Michael J. Zyphur
University of Queensland

George Sugihara
University of California San Diego

Patrick J. Laub
UNSW

Acknowledgments

Discovery Project DP200100219 and Future Fellowship FT140100629.

A different view of causality

Imagine x_t, y_t, z_t are interesting time series…

If the data is generated according to the nonlinear system:

$$x_{t+1} = \sigma(y_t - x_t)$$
$$y_{t+1} = x_t(\rho - z_t) - y_t$$
$$z_{t+1} = x_t y_t - \beta z_t$$

then $y \Rightarrow x$, both $x, z \Rightarrow y$, and both $x, y \Rightarrow z$.

Empirical Dynamic Modelling (EDM)

Create lagged embeddings

Given two time series, create E-length trajectories

$$\overleftarrow{\mathbf{x}}_t = (\text{Temp}_t, \text{Temp}_{t-1}, \dots, \text{Temp}_{t-(E-1)}) \in \mathbb{R}^E$$

and targets

$$y_t = \text{Crime}_t .$$

Note

The $\overleftarrow{\mathbf{x}}_t$'s are called *points* (on the shadow manifold).

Split the data

- $\mathcal{L} = \{(\overleftarrow{\mathbf{x}}_1, y_1), \dots, (\overleftarrow{\mathbf{x}}_n, y_n)\}$ is *library set*,
- $\mathcal{P} = \{(\overleftarrow{\mathbf{x}}_{n+1}, y_{n+1}), \dots, (\overleftarrow{\mathbf{x}}_T, y_T)\}$ is *prediction set*.

For point $\overleftarrow{\mathbf{x}}_s \in \mathcal{P}$, pretend we don't know y_s and try to predict it.

$$\forall \overleftarrow{\mathbf{x}} \in \mathcal{L} \quad \text{find} \quad d(\overleftarrow{\mathbf{x}}_s, \overleftarrow{\mathbf{x}})$$

This is computationally demanding.

Non-parametric prediction: simplex

For point $\overleftarrow{\mathbf{x}}_s \in \mathcal{P}$, find k nearest neighbours in $\mathcal{L}$.

Say, e.g., $k = 2$ and the neighbours are

$$\mathcal{N}\mathcal{N}_k = \big((\overleftarrow{\mathbf{x}}_3, y_3), (\overleftarrow{\mathbf{x}}_5, y_5)\big)$$

The *simplex method* predicts

$$\hat{y}_s = w_1 y_3 + w_2 y_5 .$$

Non-parametric prediction: S-map

Sequential Locally Weighted Global Linear Maps (S-map)

Weight the points by distance

$$w_i = \exp\{-\theta d(\overleftarrow{\mathbf{x}}_s, \overleftarrow{\mathbf{x}}_i)\}.$$

Build a local linear system

$$\widehat{y}_s = \mathbf{A}_s \overleftarrow{\mathbf{x}}_s.$$

For all $s \in \mathcal{P}$, compare $\widehat{y}_s$ to true y_s, and calculate ρ.

Convergent cross mapping

- If Temp_t causes Crime_t, then information about Temp_t is somehow embedded in Crime_t.
- By observing Crime_t, we should be able to forecast Temp_t.
- By observing more of Crime_t (more "training data"), our forecasts of Temp_t should be more accurate.

Example: Chicago crime and temperature.

Software

Stata package

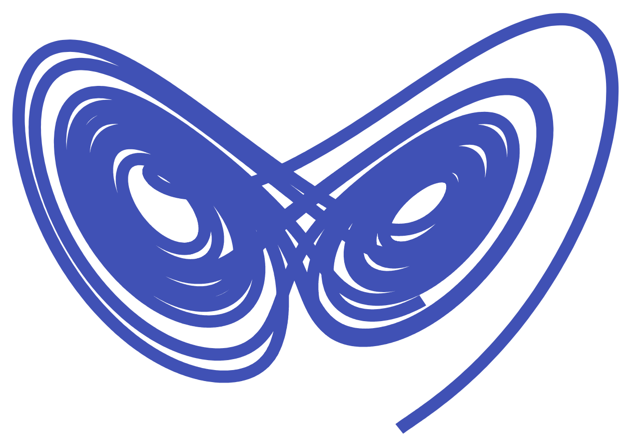
EDM Stata Package

Search

Empirical Dynamic Modeling Stata Package

Package Description

Empirical Dynamic Modeling (EDM) is a way to perform *causal analysis on time series data*. The `edm` Stata package implements a series of EDM tools, including the convergent cross-mapping algorithm.



Key features of the package:

- powered by a fast multi-threaded *C++ backend*,
- able to process panel data, a.k.a. *multispatial EDM*,
- able to handle *missing data* using new `dt` algorithms or by dropping points,
- *factor variables* can be added to the analysis,
- *multiple distance functions* available (Euclidean, Mean Absolute Error, Wasserstein),
- *GPU acceleration* available.

Table of contents

Package Description
Installation
R & Python packages
Other Resources
Authors
Citation

R package

Thanks to Rishi Dhushiyandan for his hard work on `easy_edm`.

fastEDM 0.1 Reference Articles Changelog Search for

fastEDM

The `fastEDM` R package implements a series of *Empirical Dynamic Modeling* tools that can be used for *causal analysis of time series* data.

Key features of the package:

- powered by a fast multi-threaded *C++ backend*,
- able to process panel data, a.k.a. *multispatial EDM*,
- able to handle *missing data* using new `dt` algorithms or by dropping points.

Installation

You can install the development version of fastEDM from GitHub with:

License

MIT + file LICENSE

Citation

Citing fastEDM

Developers

Patrick Laub
Author, maintainer

Jinjing Li
Author

Michael Zyphur
Author

More about authors...

Python package

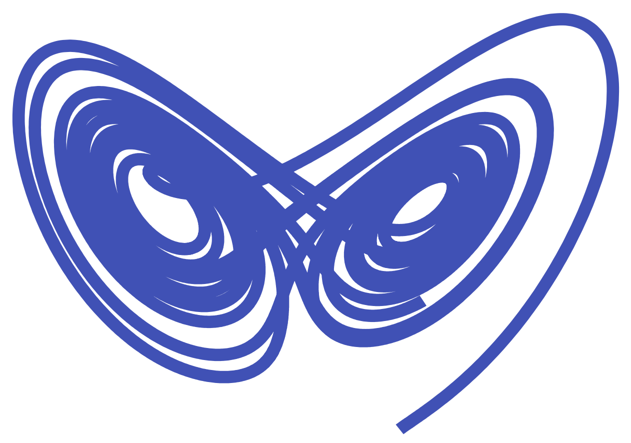
fastEDM Python Package

Search

fastEDM Python Package

Package Description

Empirical Dynamic Modeling (EDM) is a way to perform *causal analysis on time series data*. The `fastEDM` Python package implements a series of EDM tools, including the convergent cross-mapping algorithm.



Key features of the package:

- powered by a fast multi-threaded *C++ backend*,
- able to process panel data, a.k.a. *multispatial EDM*,
- able to handle *missing data* using new `dt` algorithms or by dropping points.

Installation

To install the latest version from Github using `pip` run:

Table of contents

Package Description
Installation
Example: Chicago crime levels and temperature
Stata & R packages
Other Resources
Authors
Citation

Modern engineering

- Open code (9,745 LOC) on MIT License,
- unit & integration tests (5,342 LOC),
- documentation (3,157 LOC),
- Git (1,198 commits),
- Github Actions (11 tasks),
- vectorised, microbenchmarking, ASAN, linting,
- all C++ compilers, WASM, all OSs.

Get involved!

😊 Give it a try, feedback would be very welcome.

😍 If you're talented in causal inference or programming (Stata/Mata, R, Javascript, C++, Python), we'd love contributions!

😎 "We actually also have a PhD scholarship if we have a domestic applicant who can commit like in next week" (need ≥ 85 in a 4 year degree)

Appendix

Linear/nonlinear dynamical systems

Say $\mathbf{x}_t = (x_t, y_t, z_t)$, then if:

$$\mathbf{x}_{t+1} = \mathbf{A}\mathbf{x}_t$$

we have a linear system.

$$\mathbf{x}_{t+1} = f(\mathbf{x}_t)$$

we have a nonlinear system.

> Using a term like nonlinear science is like referring to the bulk of zoology as the study of non-elephant animals. (Stanisław Ulam)

We don't fit a model for f, non-parametrically use the data. Hence the name *empirical* dynamic modelling.

Noise / unobserved variables?

Takens' theorem to the rescue, though…

> Takens' theorem is a deep mathematical result with far-reaching implications. Unfortunately, to really understand it, it requires a background in topology. (Munch et al. 2020)

Takens's theorem

From Wikipedia, the free encyclopedia

In the study of dynamical systems, a **delay embedding theorem** gives the conditions under which a chaotic dynamical system can be reconstructed from a sequence of observations of the state of a dynamical system. The reconstruction preserves the properties of the dynamical system that do not change under smooth coordinate changes (i.e., diffeomorphisms), but it does not preserve the geometric shape of structures in phase space.

Contents [hide]

1 Simplified, slightly inaccurate version
2 See also
3 References
4 Further reading
5 External links

Simplified, slightly inaccurate version [edit]

Suppose the d-dimensional state vector x_t evolves according to an unknown but continuous and (crucially) deterministic dynamic. Suppose, too, that the one-dimensional observable y is a smooth function of x, and "coupled" to all the components of x. Now at any time we can look not just at the present measurement $y(t)$, but also at observations made at times removed from us by multiples of some lag $\tau : y_{t+\tau}, y_{t+2\tau}$, etc. If we use k lags, we have a k-dimensional vector. One might expect that, as the number of lags is increased, the motion in the lagged space will become more and more predictable, and perhaps in the limit $k \to \infty$ would become deterministic. In fact, the dynamics of the lagged vectors become deterministic at a finite dimension; not only that, but the deterministic dynamics are completely equivalent to those of the original state space (More exactly, they are related by a smooth, invertible change of coordinates, or diffeomorphism.) The magic embedding dimension k is at most $2d + 1$, and often less.[1]

Source: Munch et al. (2020), Frequently asked questions about nonlinear dynamics and empirical dynamic modelling, ICES Journal of Marine Science.