Modeling high-dimensional natural inflows for stochastic-dynamic optimization

Workshop on Hydro Scheduling in Competitive Electricity Markets
Trondheim (Norway), September 17-18, 2015

Nils Löhndorf
Assistant Professor
WU Vienna University of Economics and Business, Austria

Andreas Eichhorn
Portfolio Management
Verbund Trading, Austria

WIRTSCHAFTS UNIVERSITÄT WIEN VIENNA UNIVERSITY OF ECONOMICS AND BUSINESS

Stochastic Dynamic Optimization

- Dynamic decision problem with T time stages

  $\xi = (\xi_1, \dots, \xi_T)$ with $\xi_t(\omega) \in \mathbb{R}^m$ and $\xi^t = (\xi_1, \dots, \xi_t)$

  $x = (x_1, \dots, x_T)$ decisions with $x^t = (x_1, \dots, x_t)$ and $x_t \in \mathcal{X}_t$

  x_t measurable w.r.t. $\sigma(\xi^t)$

- Model formulation using value / cost-to-go functions

$$V_T(x^{T-1}, \xi^T) = \max_{x_T \in \mathcal{X}_T} R_T(x^T, \xi^T)$$

$$V_t(x^{t-1}, \xi^t) = \max_{x_t \in \mathcal{X}_t} R_t(x^t, \xi^t) + \mathbb{E}\left[V_{t+1}(x^t, \xi^{t+1}) | \xi^t\right], \ \forall t : 1 \leq t < T$$

- Numerical solution

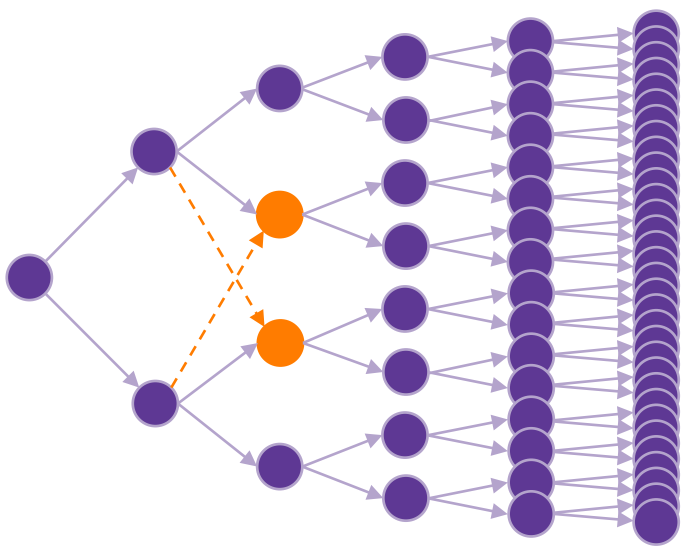
  Integrals $\mathbb{E}\left[V_{t+1}(x^t, \xi^{t+1}) | \xi^t\right]$ have to be calculated

  Discretization of ξ required for numerical solutions

From Trees to Lattices

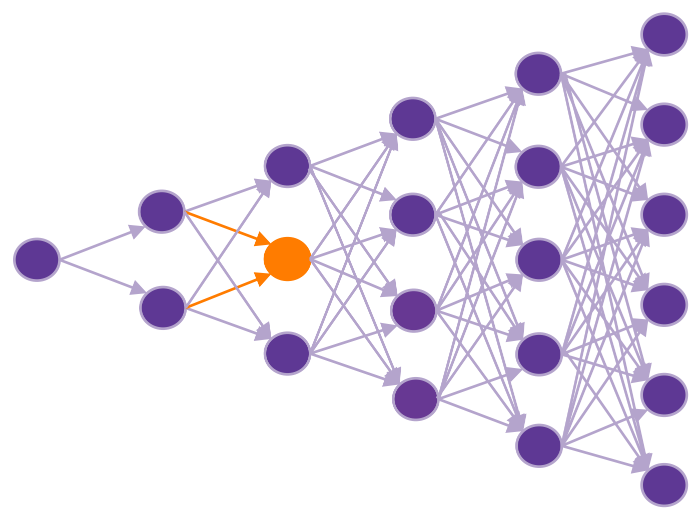
Scenario Generation for Stochastic Optimization

Scenario Tree



State-time-*history* graph with
63 nodes and 32 scenarios

Scenario *Lattice*



State-time graph with
21 nodes and 720 scenarios

Scenario Lattices
A Compressed Scenario Representation

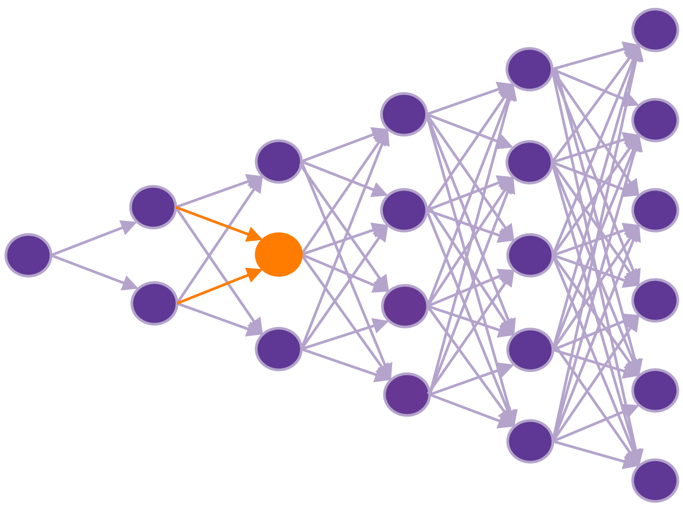
WIRTSCHAFTS UNIVERSITÄT WIEN VIENNA UNIVERSITY OF ECONOMICS AND BUSINESS

- Lattices carry no information about the history of the process
- Suitable for Markov processes where

  $$P(\xi_{t+1} \in A|\xi^t) = P(\xi_{t+1} \in A|\xi_t)$$

- Covers all state space type stochastic models:
  - Autoregressive models
  - Dynamic factor models
  - Exponential smoothing
  - Hidden Markov models
  - Stochastic volatility models

Scenario *Lattice*



Stochastic Optimization on Lattices
From optimal decisions to value functions

- Decisions dependent on path leading to node *n*

$$x_n = x_n(x^{n-1}, \xi^t)$$

- Past decisions aggregate into a *resource* state (inventory, reservoir, …)

$$S_t(x_1, \xi_1, \ldots, x_{t-1}, \xi_{t-1}) = S_t(S_{t-1}, x_{t-1}, \xi_{t-1})$$

- Decisions are based on current state of the 'world'

$$x_t(x_1, \xi_1, \ldots, x_{t-1}, \xi_{t-1}, \xi_t) = x_t(S_t, \xi_t)$$

- *Markov Decision Process (MDP)* if the random process is memoryless and the value depends only on the current state

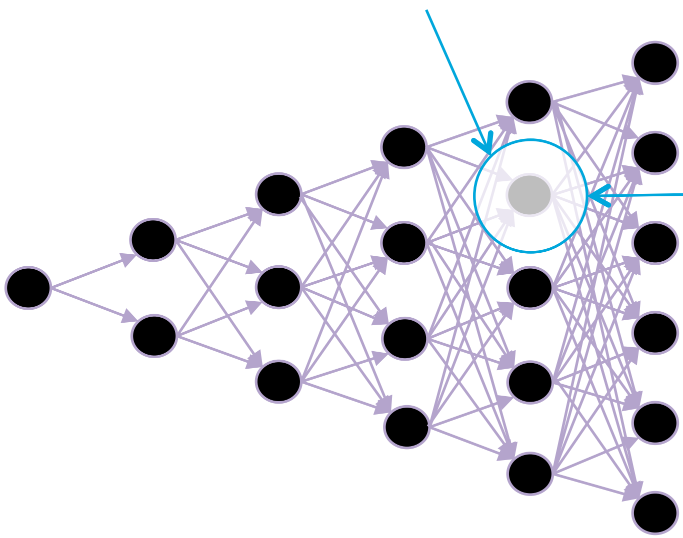
$$\mathbb{E}\left[V_t(x^{t-1}, \xi^t) | \xi^{t-1}\right] = \mathbb{E}\left[V_t(S_t, \xi_t) | \xi_{t-1}\right]$$

Approximate Dual Dynamic Programming

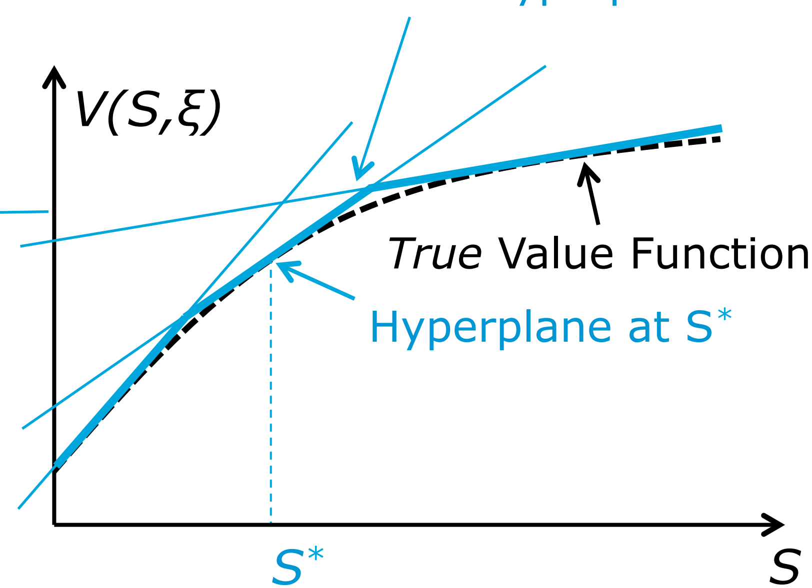
A solution strategy using scenario lattices

1. Construct a scenario lattice from a stochastic process
2. Find an approximate value function for each node

Each node of the lattice holds a value function



Approximate the value function as minimum of a set of hyperplanes



How Do We Fit a Lattice From Data?

- Parametric approach
  1. Estimate a Markovian time series process
  2. Reduce the continuous Markovian process to a lattice
  3. Determine an optimal policy using ADDP
  4. Evaluate the policy on the original time series model
- Data-driven approach
  1. Estimate a lattice directly from data
  2. Determine an optimal policy using ADDP

Skip scenario reduction

+

No policy validation needed

Multivariate Time Series Models

- Time series models with many variables
  - Vector autoregressive (VAR) models
  - Vector error correction (VEC) models
  - Multifactor models of state space type
- Dynamic factor model
  - Markovian time series model
  - Observations as linear combination of a few hidden factors
  - Less parameters through dimensionality reduction

Dynamic Factor Model

t: time index
F_t: vector of factors
A: transition matrix
X_t: vector of observations
V: dynamic factor loadings
B: parameter matrix of exogenous predictors
Z_t: exogenous predictors
u_t, v_t: error term

$$F_t = AF_{t-1} + u_t$$

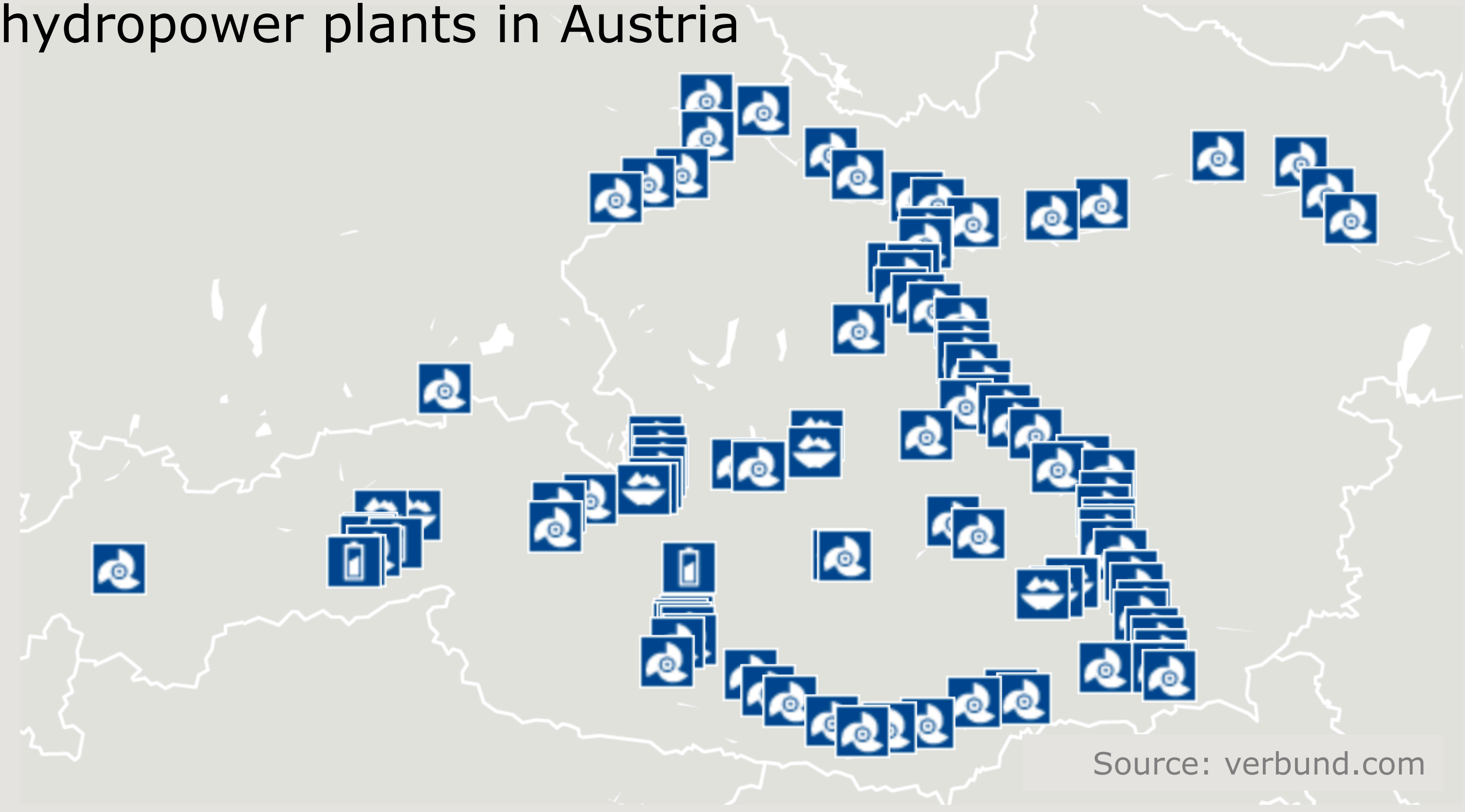
$$X_t = VF_t + BZ_t + v_t$$

Dynamic Factor Lattice

- Building blocks
  - Represent factor space by set of discrete factors $\bar{F}$
  - Replace linear state transition equation with transition matrices $P_t(\bar{F}_t|\bar{F}_{t-1})$
- Learning a lattice from data
  1. Choose optimal quantizers of the data at stage *t*
  2. Count transitions between observations in the neighborhood of quantizers at successive stages
  3. We use a moving time window of 30 days of observed transitions to obtain a large enough sample

Verbund Case Study

- Inflow data
  - Historical data from 1990 to 2012
  - Daily incremental inflows of 50 rivers
- Verbund hydropower plants in Austria

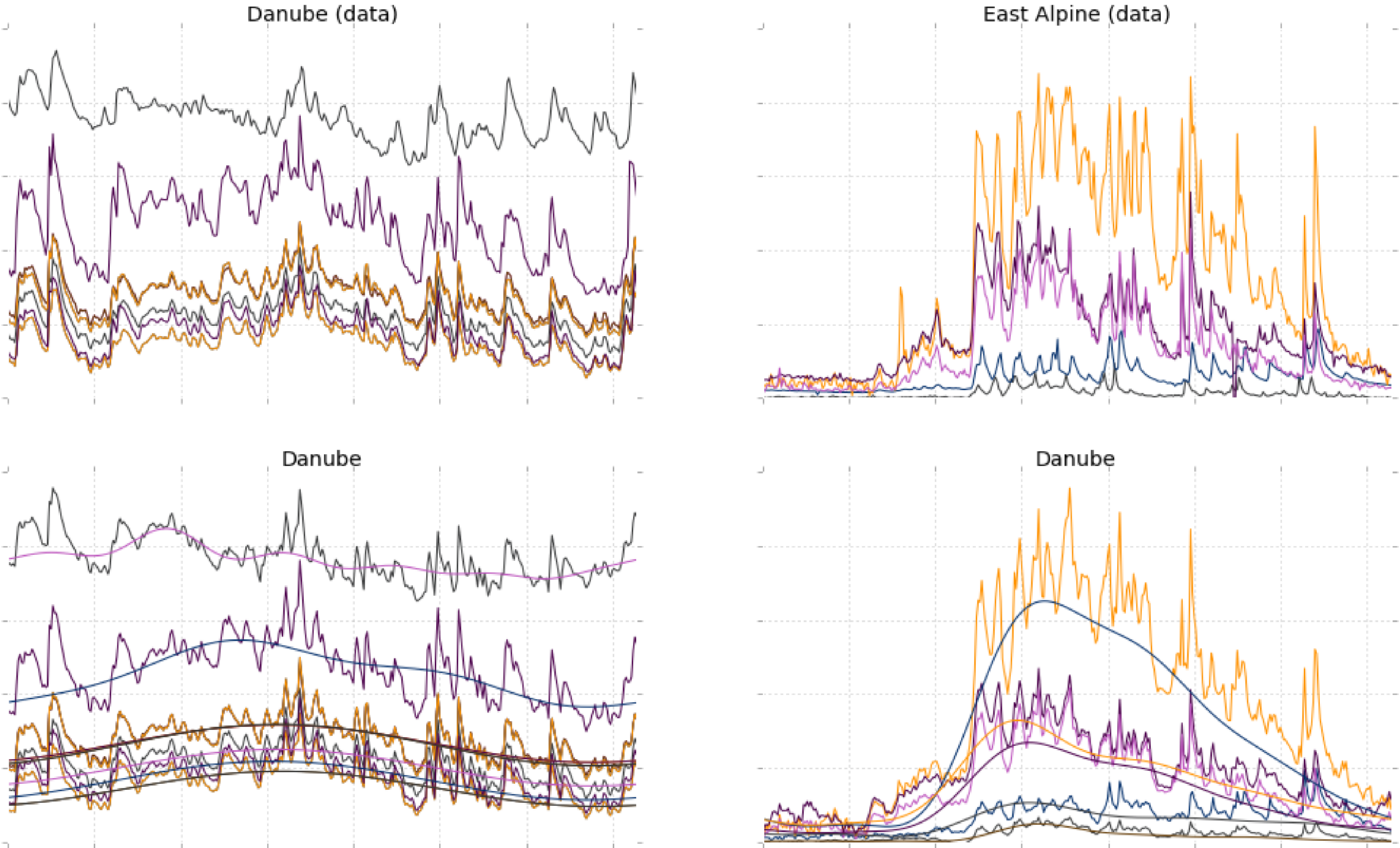


Source: verbund.com

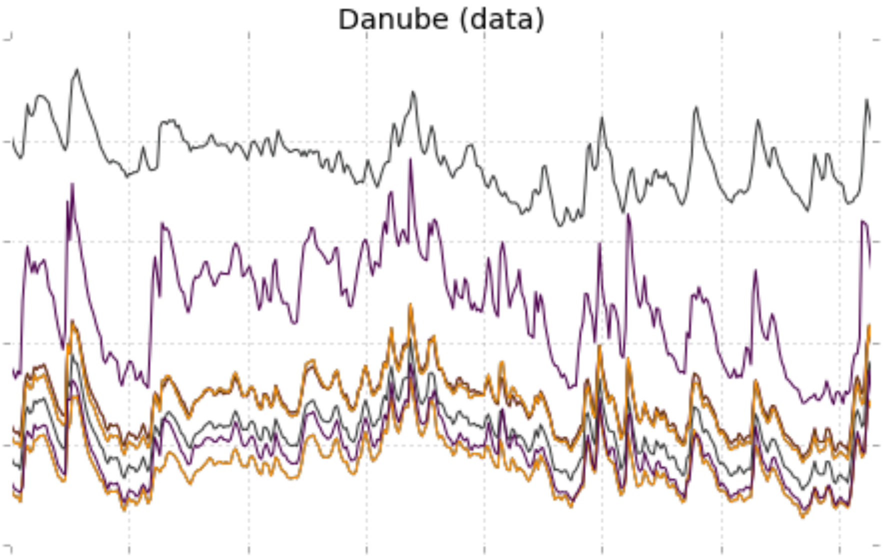
Methodology

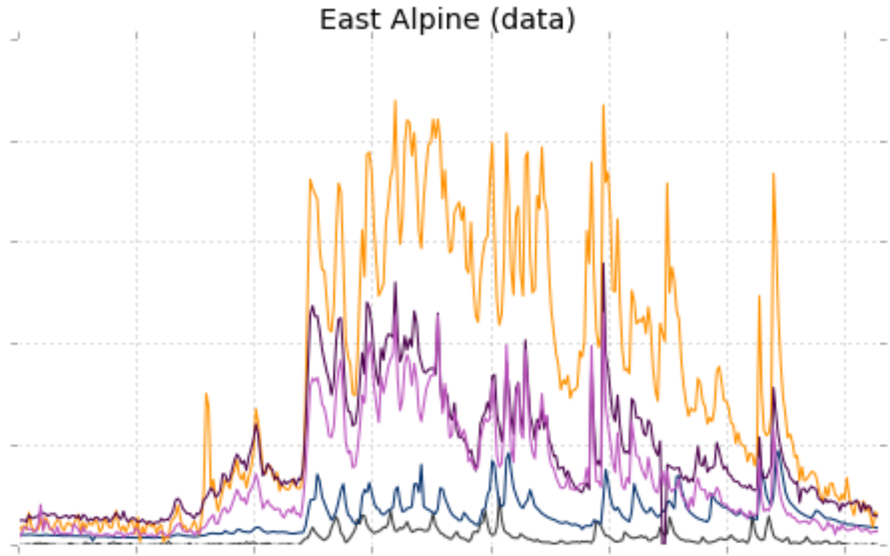
1. Transformation
   - Negative inflows prohibit Box-Cox transformation
   - Inverse hyperbolic sine transformation
2. Time series regression
   - Seasonal component as Fourier series
   - Model selection using BIC
3. Missing values
   - Singular value decomposition
   - Threshold value selected via cross-validation
4. Dynamic Factor Model
   - Static approach using PCA
   - 3 factors capture 75% of the variance
5. Lattice
   - 50 states + 364 transition matrices

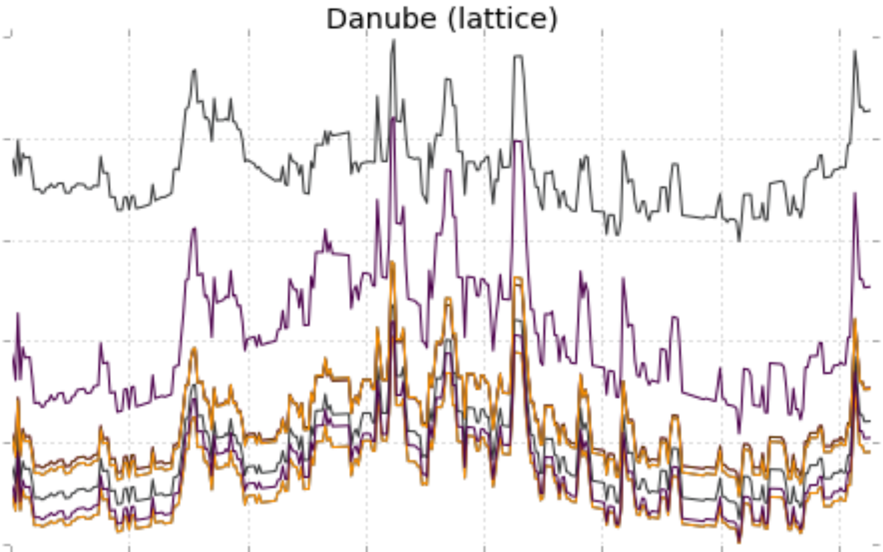
Original vs Reduced Time Series

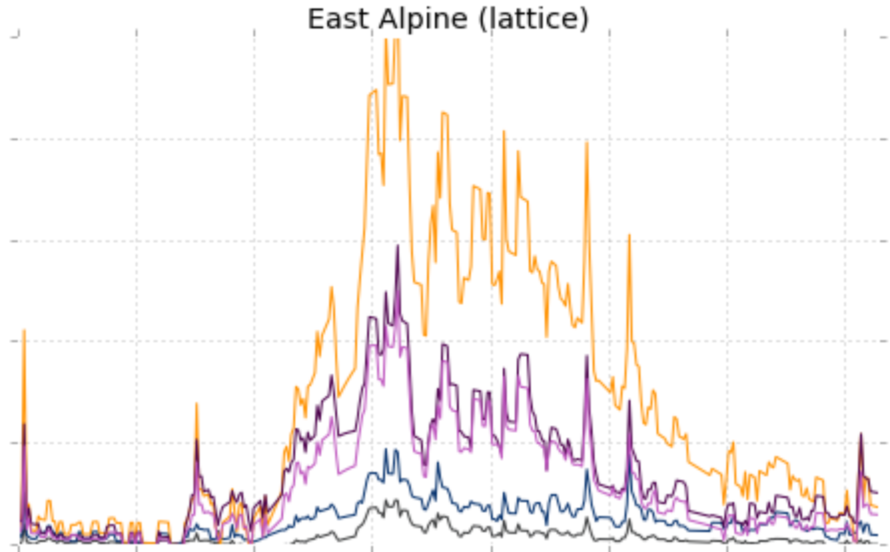


Original vs Simulated Time Series

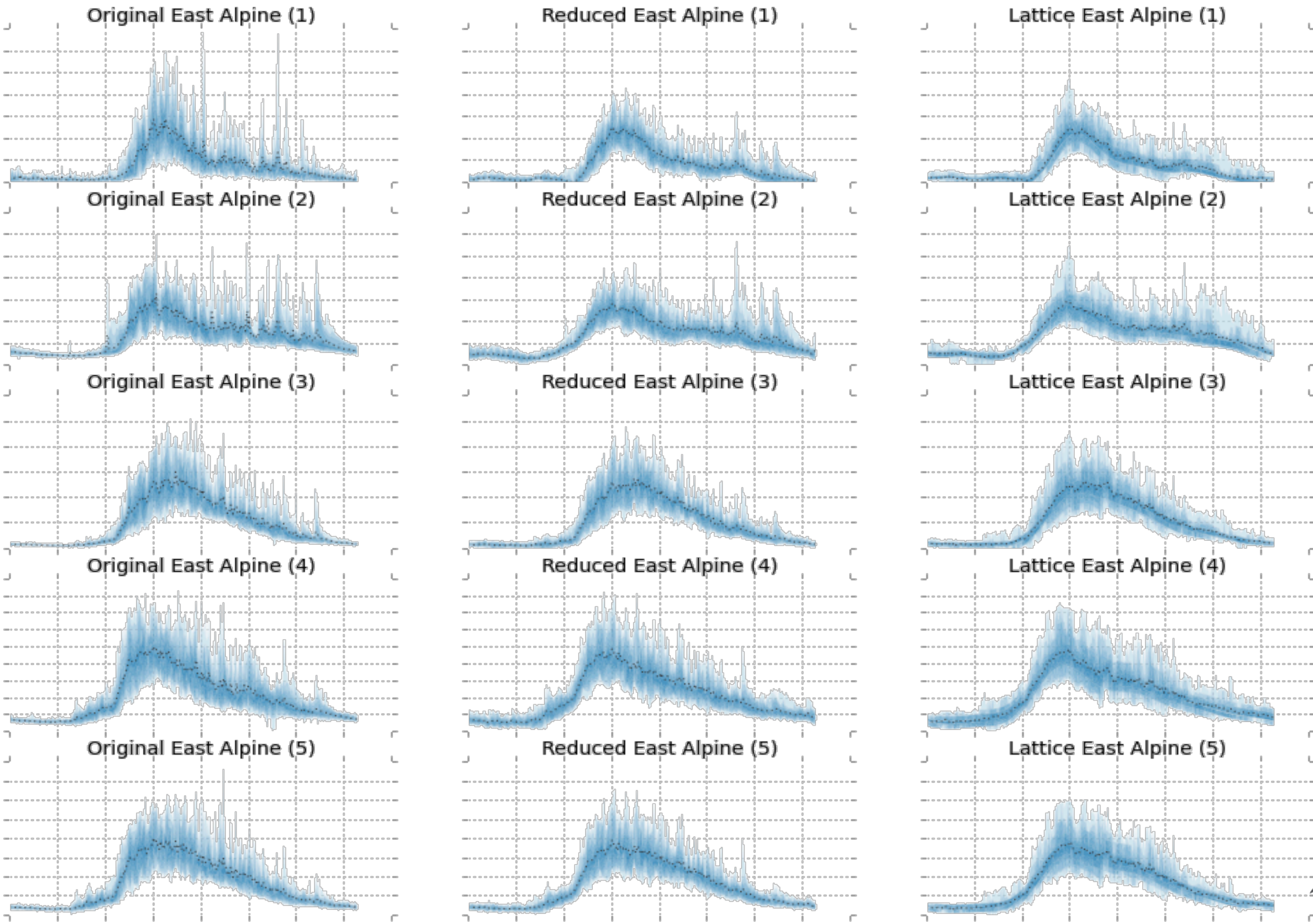






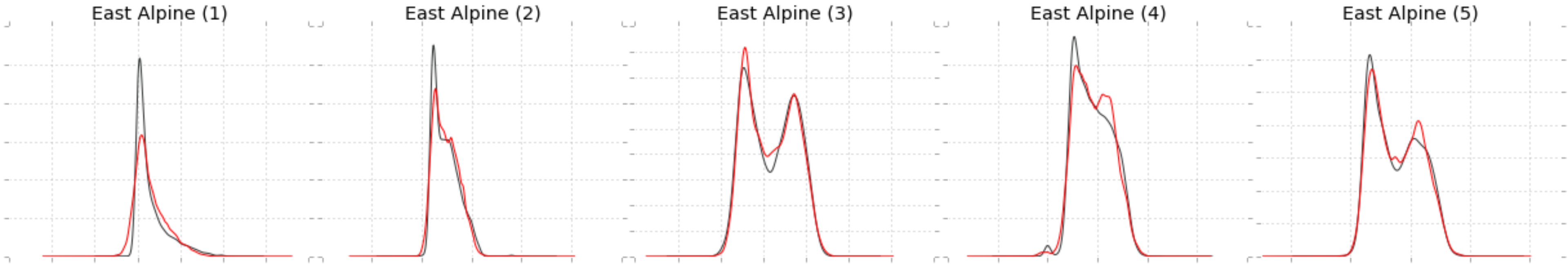


Data → Factors → Lattice

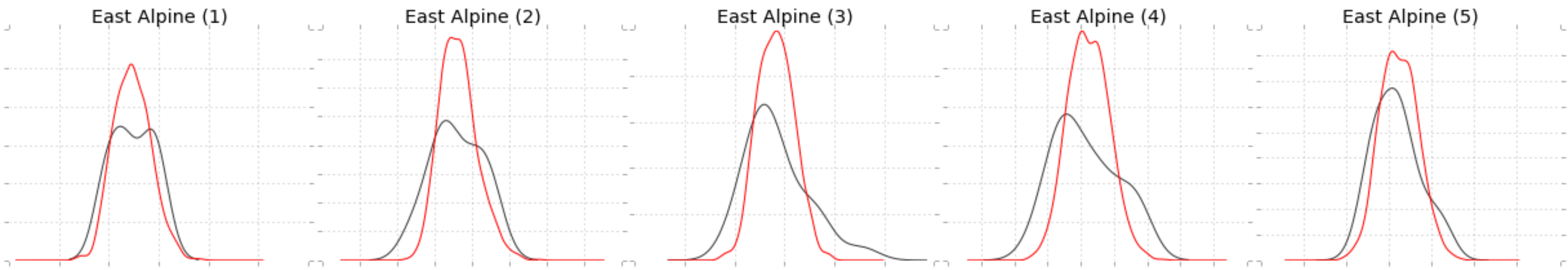


Kernel Density Estimation

Mean daily inflows



Mean annual inflows



– original data – lattice simulation

Conclusion

- Results
  - A small number of discrete states is sufficient to explain a high-dimensional inflow process.
  - Factors achieve longitudinal smoothing which helps to extract long-range information
- Outlook
  - Semi-parametric extensions to overcome sparse state transition samples
- Future Work
  - How do we measure the goodness of fit of a lattice?

Quantego QUASAR
A general-purpose stochastic optimizer