Advanced Time Series

Lecture 6:
Representation learning

Gleb Ivashkevich

Today

T. t. e. $\longrightarrow$ representation learning:

- survival analysis modeling and DL formulation
- representation learning setup
- time-lagged autoencoder
- VAMPnets
- wrap-up

Survival analysis and predictive maintenance

Survival analysis

Concept 1:

- **survival function:** probability of surviving past t

$$S(t) = P(\underbrace{T}_{\text{time of failure}} > t)$$

Survival analysis

Concept 2:

- **hazard function:** conditioned event rate

$$\lambda(t) = -\frac{S'(t)}{S(t)}$$

Hazard function deciphered

Concept 2:

- **hazard function:** conditioned event rate

$$\lambda(t) = -\frac{P(T>t+dt)-P(T>t)}{S(t)dt}$$

Hazard function deciphered

Concept 2:

- **hazard function:** conditioned event rate

$$\lambda(t) = -\frac{P(T > t + dt) - P(T > t)}{S(t)dt}$$

Proportional hazards

Concept 3:

- **proportional** hazards model (Cox regression)

$$\lambda(t) = \lambda_0(t)\exp(a_i X_i)$$

covariates

Hazards and s. f.

Concept 4:

- **cumulative** hazard

$$\Lambda(t) = \int_0^t \lambda(\tau) d\tau \rightarrow S(t) = \exp(-\Lambda(t))$$

cumulative hazard

Survival models

non-parametric	semi-parametric	parametric	ML
Kaplan-Meier Nelson-Aalen	Cox PH	AFT models	survival trees, etc.

Survival data

Covariates X_i^k: a vector (i) per object (k)

Lifespan T^k

Event was observed? C^k

Notes:

- one vector of covariates **for entire lifespan**
- some events are **censored** (object **"died"**, but for a different reason)

Survival data: extensions

Covariates **may vary with time** $X_i^k(t)$

For example:

- each object is measured multiple times

Cox regression: likelihood

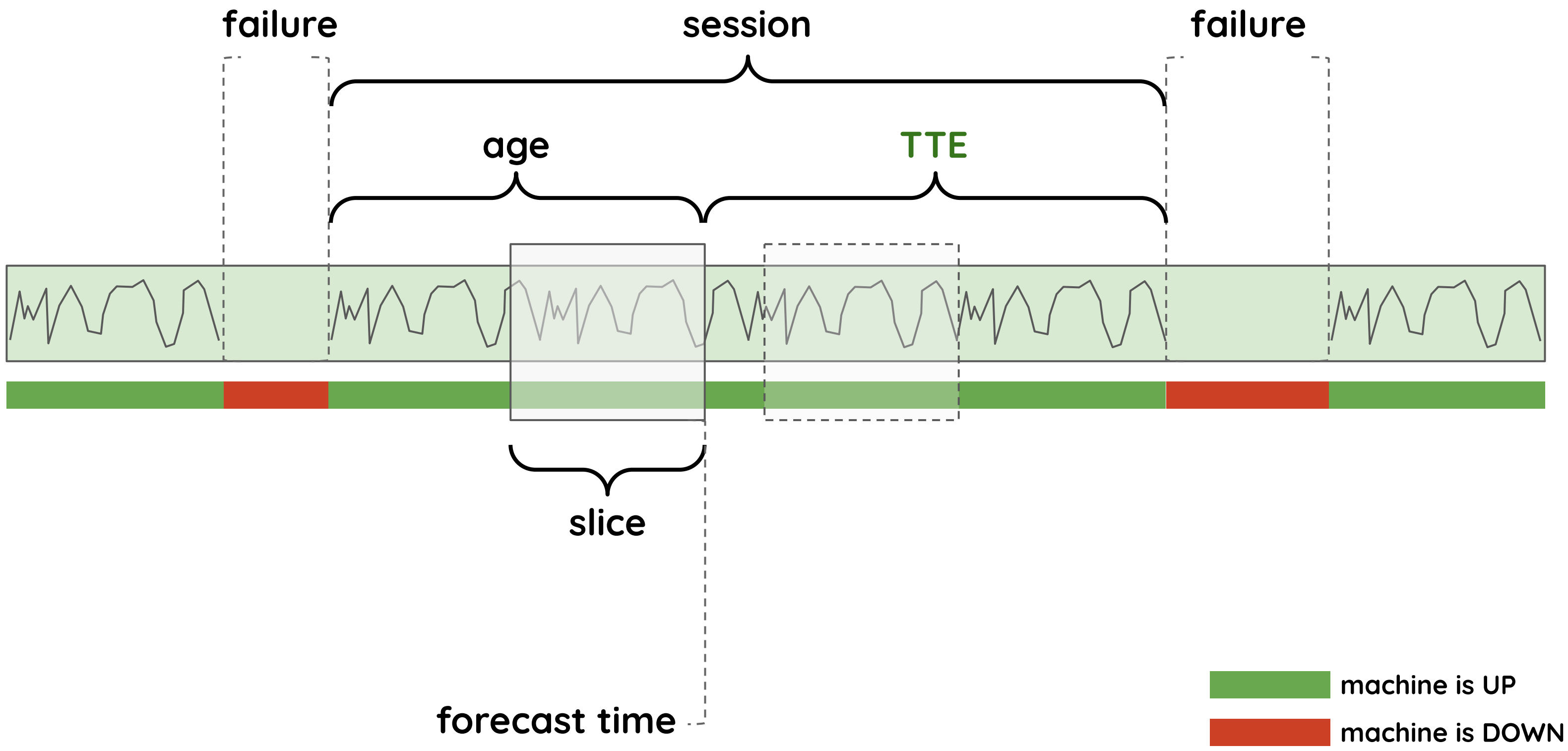
How can we train Cox PH?

Full likelihood **is not specified** (because of baseline hazard). Partial likelihood (for individual object and no ties):

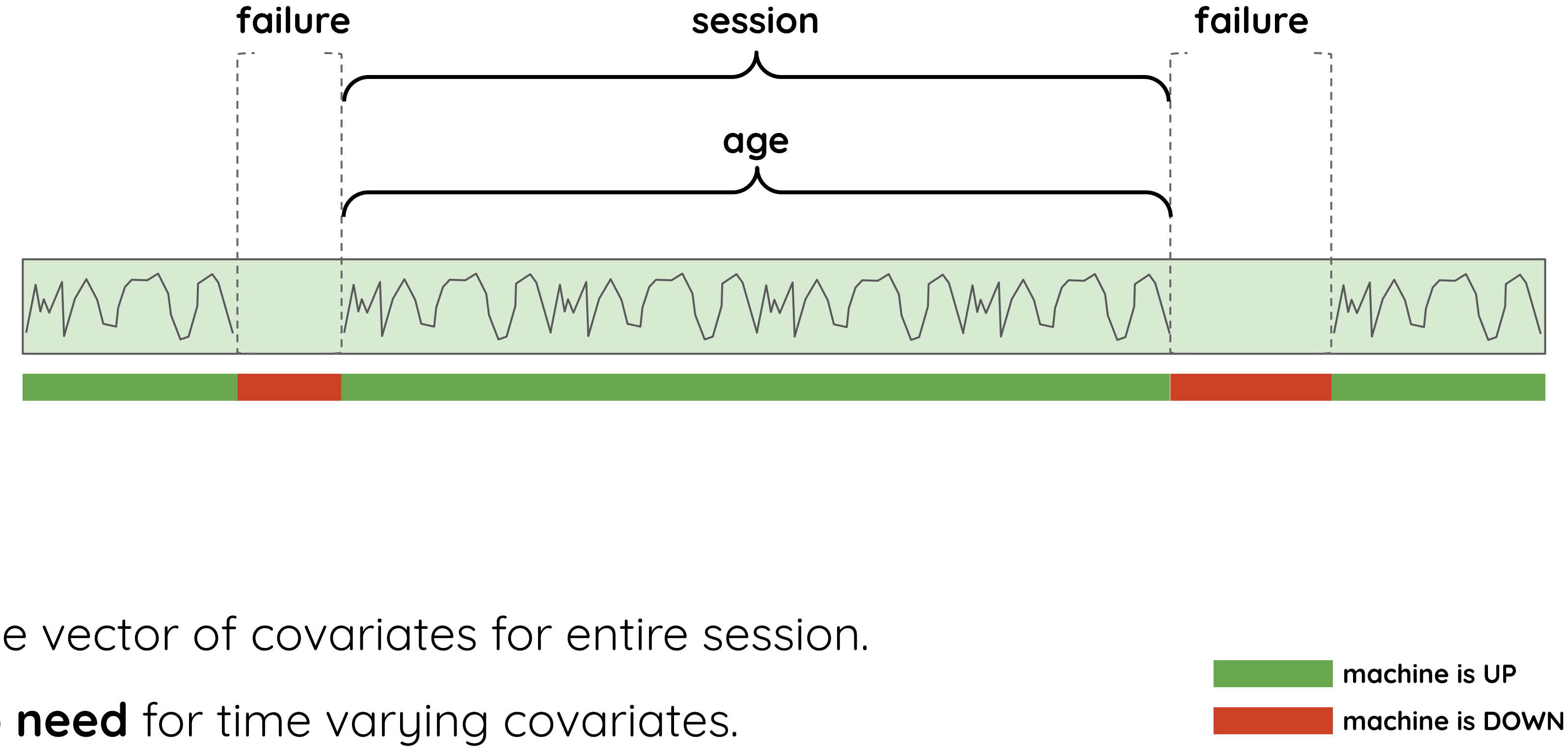
$$L_i = \frac{\lambda(t^i|X^i)}{\sum_{t^j \geq t^i} \lambda(t^i|X^j)} = \frac{\exp\left(a_\alpha X^i_\alpha\right)}{\sum_{t^j \geq t^i} \exp\left(a_\alpha X^j_\alpha\right)}$$

$$L = \prod L_i$$

PdM data setup: slices



PdM data setup: sessions



One vector of covariates for entire session.

No need for time varying covariates.

Realistic PdM

Some considerations:

- model each type of failure **separately**
  (slices/sessions ended with a different failure are censored)
- session-based analysis for **post-mortem analysis**
- try session-based models for real-time predictions with **expanding windows** (may work for frequent failures)

DL and survival analysis

Some considerations:

- (partial)likelihoods are known for semi-parametric and parametric models
- encode time series: encoder
- push them into the appropriate model
- use Cox or something else as a baseline

DL and survival analysis

For example, for CoxPH:

- if you’re lucky, you may train encodings, which work well in PH model:

$$\lambda(t) = \lambda_0(t) \exp(a_i \underbrace{X_i}_{\text{encoding}})$$

DL and survival analysis

Non-specific to time series:

- [DeepSurv]
- [DeepHit]

Representation learning

Representations for t.s.

When:

- highly dimensional time series with complex patterns
- barely interpretable

Why:

- denser
- hopefully, provide some insights into structure
- simplify forecasting, classification and t.t.e.: substitute for pre-training

Representations for t.s.

Applications:

- manufacturing data
- molecular dynamics data
- various medical data

Naive: PCA

When:

- simple linear dependencies between(pointwise!) covariates

Why not:

- you never know if it’s linear or not
- temporal information is not used (neither short-term, nor long-term)

Reasonable: TICA

Time-lagged **I**ndependent **C**omponent **A**nalysis:
temporal extension of PCA

How:

- instead of solving eigenvalue problem for covariance matrix, solve it for auto-covariance matrix:

$$C_{ij}(\tau) = \frac{1}{T-\tau-1} \sum_{t=1}^{t=T-\tau} x_i(t) x_j(t+\tau)$$

Reasonable: TICA

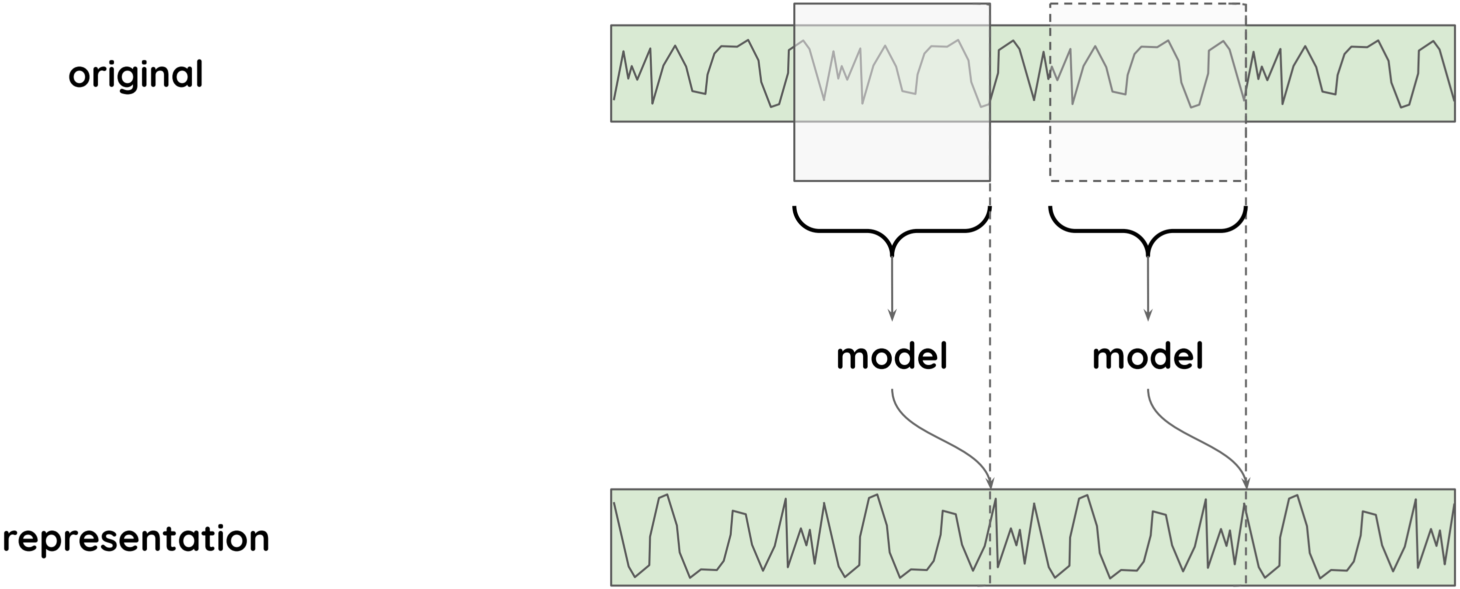
Pros:

- temporal information is partially included
- more meaningful components

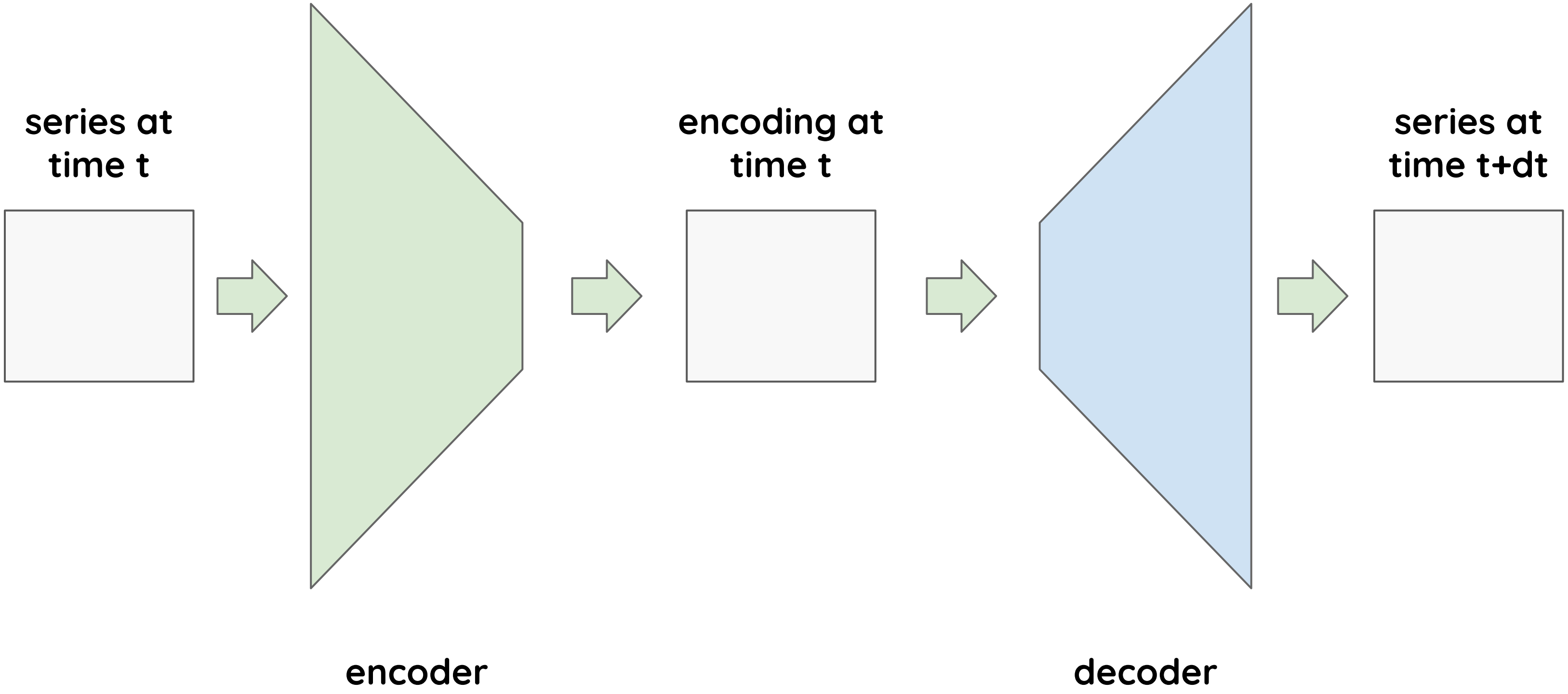
Cons:

- still linear
- no patterns are accounted for: it's pointwise

Windowed representation



Time-lagged autoencoder



Time-lagged autoencoder

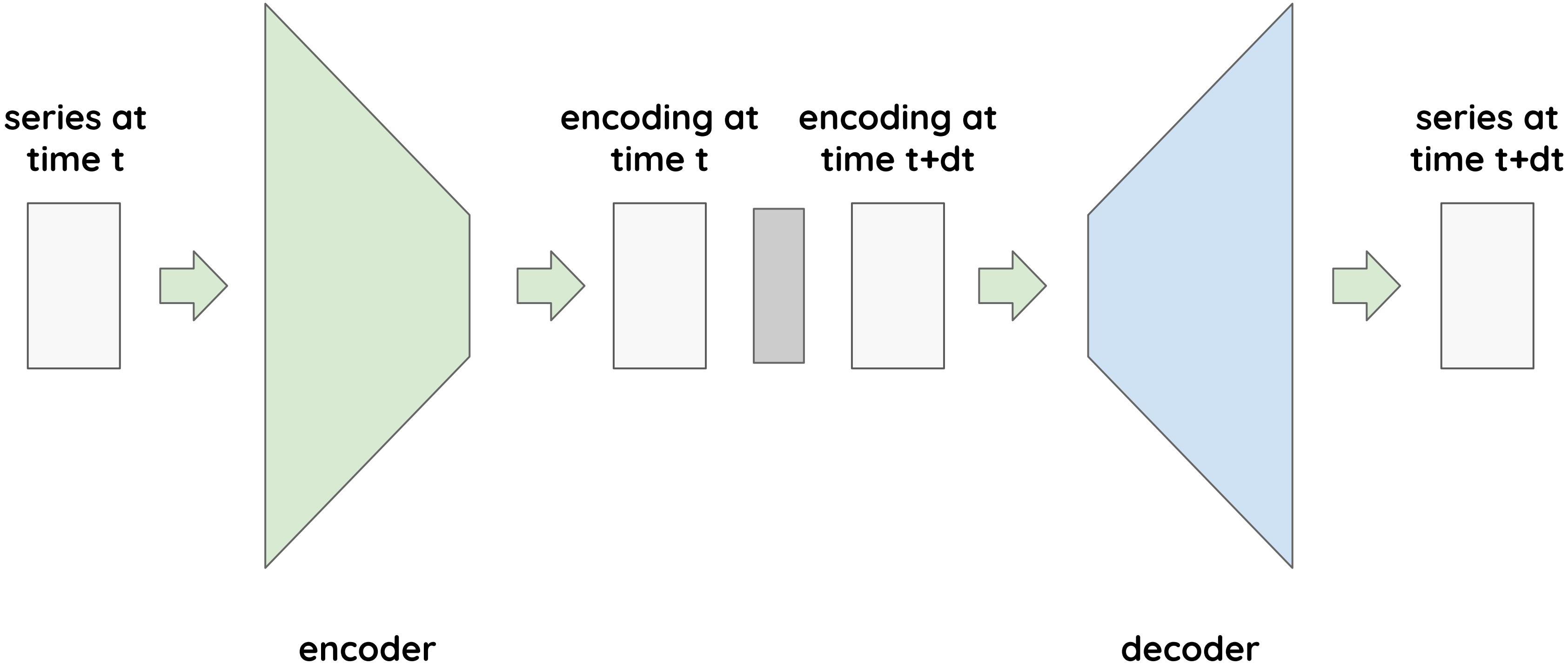
Pros:

- conceptually simple
- can contain any blocks needed (CNN, RNN, etc.)
- trained with MSE

Cons:

- may not find problem-specific representation
- no fundamental guarantees

TLA with propagator



TLA with propagator

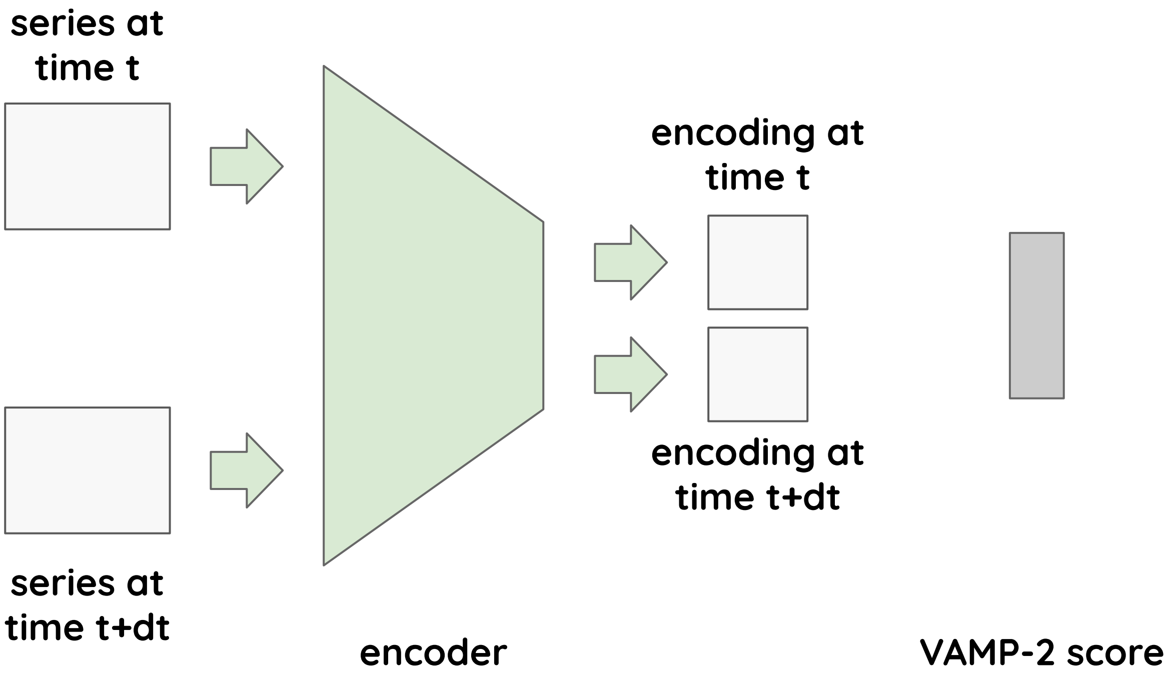
Pros:

- attempts to estimate (non-linear) dynamics

Cons:

- can be hard to train
- must be trained with several lags (as it's impossible to separate propagator from encoder and decoder)
- still ad hoc

VAMPnets


series at
time t
series at
time t+dt
encoder
encoding at
time t
encoding at
time t+dt
VAMP-2 score

VAMPnets

Pros:

- fundamentally validated (Koopman operator, etc.)
- provides hierarchy of relaxation times
- filters noisy, uncorrelated components

Cons:

- can be hard to train
- selection of parameters may be problematic

VAMPnets

VAMP-2 score:

- covariance matrices are calculated over transformed coordinates
- can be optimized directly

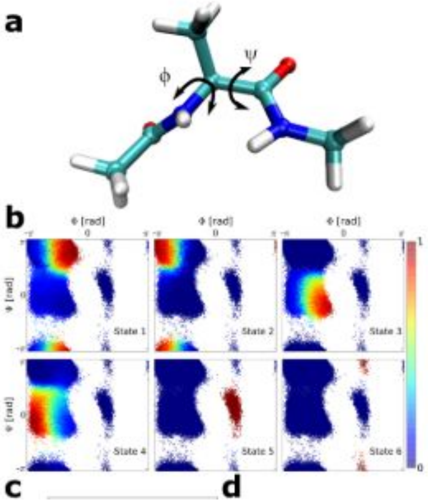
$$R = ||C_{00}^{-1/2} C_{0\tau} C_{\tau\tau}^{-1/2}||$$

[VAMPnets: Deep learning of molecular kinetics]

VAMPnets

Molecular dynamics:

- collective variables are important for modeling



Wrap-up

What we've learned

- **forecasting:** simple ED, probabilistic, transformers
- **classification**
- **survival analysis**
- **some representations learning**

Next:

- generative models, etc.

questions?