Causal Inference in Non-linear Time-series using Deep Networks and Knockoff Counterfactuals

Махин Артем, 417 группа

https://arxiv.org/pdf/2109.10817.pdf

18.10.20201

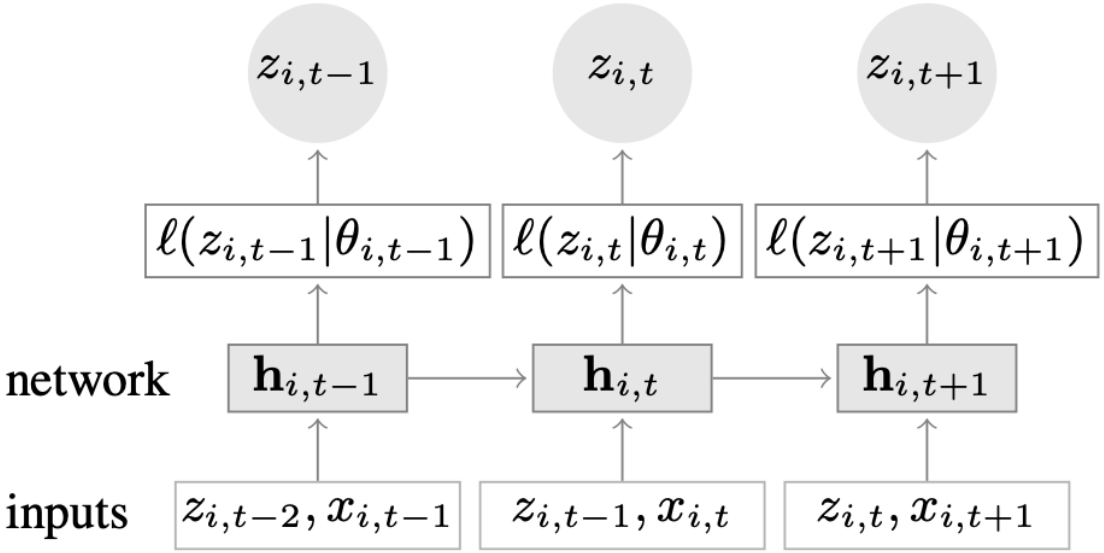
Causal Inference

Причинно-следственный вывод

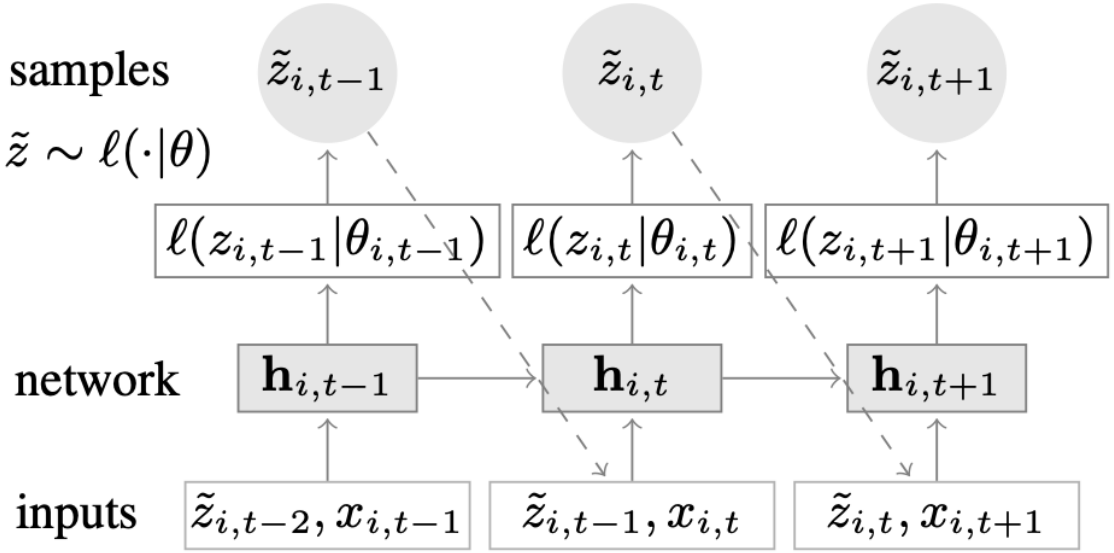
При причинном выводе человек делает вывод о том, что что-то является или может быть причиной чего-то еще.

Deep AR

Amazon, 2019



Train



Forecast

Deep AR

Amazon, 2019

Максимизируем логарифм правдоподобия: $$\mathcal{L} = \sum_{i=1}^{N} \sum_{t=t_0}^{T} \log \ell(z_{i,t} | \theta(\mathbf{h}_{i,t})) \, .$$

Deep AR

Amazon, 2019

Максимизируем логарифм правдоподобия: $$\mathcal{L} = \sum_{i=1}^{N} \sum_{t=t_0}^{T} \log \ell(z_{i,t}|\theta(\mathbf{h}_{i,t})) \,.$$

Пример: $\theta = (\mu, \sigma)$

$$\ell_{\mathrm{G}}(z|\mu,\sigma) = (2\pi\sigma^2)^{-\frac{1}{2}} \exp(-(z-\mu)^2/(2\sigma^2))$$

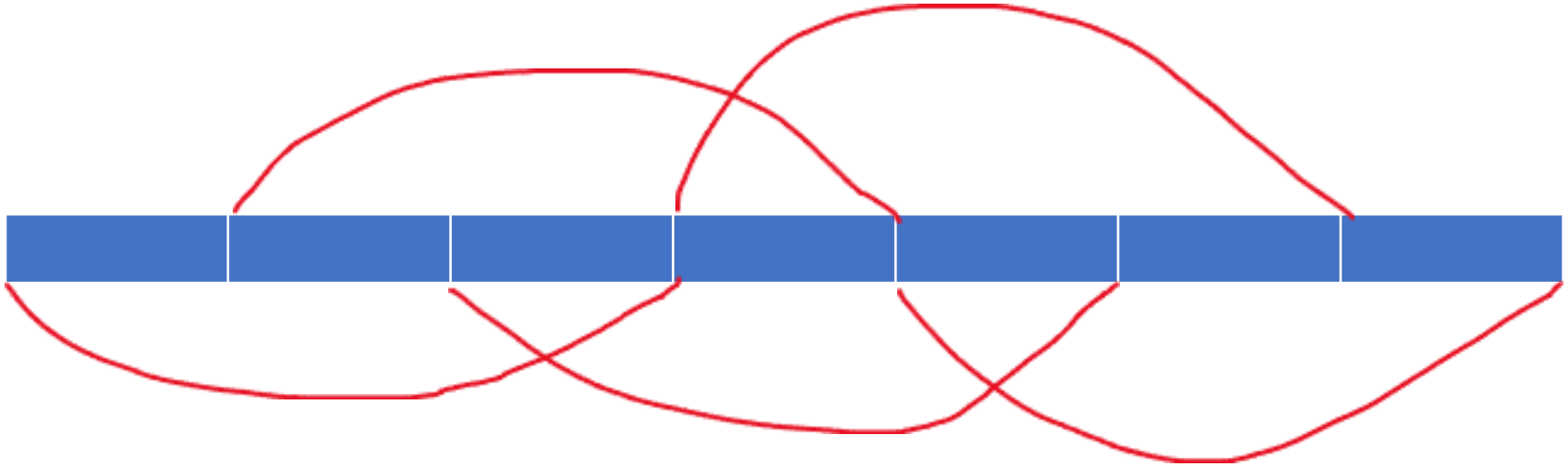
$$\mu(\mathbf{h}_{i,t}) = \mathbf{w}_\mu^T \mathbf{h}_{i,t} + b_\mu \quad \text{and} \quad \sigma(\mathbf{h}_{i,t}) = \log(1 + \exp(\mathbf{w}_\sigma^T \mathbf{h}_{i,t} + b_\sigma))$$

Deep AR

Особенности обучения

Масштабирование: $\nu_i = 1 + \frac{1}{t0} \sum_{t=1}^{t0} z_{i,t}$

Обучение окнами:



Sequence to Sequence Encoder-Decoder

Deep AR

Свойства

1

DeepAR эффективен при прогнозировании сезонных зависимостей с минимальной настройкой

2

Deep AR делает вероятностный прогноз

3

DeepAR может использовать ряды с небольшой историей

4

DeepAR поддерживает широкий спектр функций правдоподобия

Knockoffs

2015

http://dx.doi.org/10.1214/15-AOS1337

$$Z = (Z_1, \ldots, Z_n) \qquad \widetilde{Z} = (\widetilde{Z}_1, \ldots, \widetilde{Z}_n)$$

$$(Z, \widetilde{Z})_{swap(\mathbb{A})} \stackrel{d}{=} (\widetilde{Z}, Z) \quad \text{для любого} \quad \mathbb{A} \subseteq 1, \ldots, n$$

Knockoffs

2015

http://dx.doi.org/10.1214/15-AOS1337

$$Z = (Z_1, \ldots, Z_n) \qquad \widetilde{Z} = (\widetilde{Z}_1, \ldots, \widetilde{Z}_n)$$

$$(Z, \widetilde{Z})_{swap(\mathbb{A})} \stackrel{d}{=} (\widetilde{Z}, Z) \quad \text{для любого} \quad \mathbb{A} \subseteq 1, \ldots, n$$

Пример: $P_Z = \mathcal{N}_n(\mathbf{0}_n, \Sigma)$

$$P_{\widetilde{Z}|Z}(.|\mathbf{Z}_{i,*}) = \mathcal{N}_n((\mathbf{I}_n - S\Sigma^{-1})\mathbf{Z}_{i,*}, 2S - S\Sigma^{-1}S)$$

для любой фиксированной матрицы S, удовлетворяющей $0 \leq S \leq 2\Sigma$.

Causal effect estimation

$z_{i,t}, t = 1, \ldots, r, i = 1, \ldots, N$ реализации длины r процессов $Z_i, i = 1, \ldots, N$

$$\mathrm{MAPE} = \frac{1}{r} \sum_{t=1}^{r} \frac{\mid z_{i,t} - \hat{z}_{i,t} \mid}{\mid z_{i,t} \mid}$$

Causal significance score: $\mathrm{CSS}_{i \to j} = \ln \frac{\mathrm{MAPE}_j^i}{\mathrm{MAPE}_j}$

Чем заменять?

- *Knockoffs (DeepKnockoffs)*
- *Mean* $\overline{z}_i = \frac{1}{r}\sum_{t=1}^{r} z_{i,t}$
- *Out-of-distribution*

VAR GC

vector autoregressive Granger Causality

$z_{i,t}, t = 1, \ldots, r, i = 1, \ldots, N$ реализации длины r процессов $Z_i, i = 1, \ldots, N$

$$\begin{bmatrix} z_{1,t} \\ \vdots \\ z_{N,t} \end{bmatrix} = \sum_{m=1}^{p} A_m \begin{bmatrix} z_{1,t-m} \\ \vdots \\ z_{N,t-m} \end{bmatrix} + \begin{bmatrix} \epsilon_1(t) \\ \vdots \\ \epsilon_N(t) \end{bmatrix}$$

VAR GC

vector autoregressive Granger Causality

$z_{i,t}, t = 1, \ldots, r, i = 1, \ldots, N$ реализации длины r процессов $Z_i, i = 1, \ldots, N$

$$\begin{bmatrix} z_{1,t} \\ \vdots \\ z_{N,t} \end{bmatrix} = \sum_{m=1}^{p} A_m \begin{bmatrix} z_{1,t-m} \\ \vdots \\ z_{N,t-m} \end{bmatrix} + \begin{bmatrix} \epsilon_1(t) \\ \vdots \\ \epsilon_N(t) \end{bmatrix}$$

$\sum_j (\epsilon_j, z_j)$

$\sum_j^{i-} (\epsilon_j, z_j)$

VAR-GC (z_i, z_j): $\gamma_{i \to j} = \ln \frac{|\sum_j^{i-}|}{|\sum_j|}$

Эксперименты

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}}$$

$$\text{F-score} = \frac{\text{TP}}{\text{TP} + 0.5(\text{FP} + \text{FN})}$$

Эксперименты

$$\text{FPR} = \frac{\text{FP}}{\text{FP + TN}}$$

$$\text{F-score} = \frac{\text{TP}}{\text{TP} + 0.5(\text{FP + FN})}$$

$$\begin{aligned}
X_1(t) &= \mathcal{N}(0,\,1) + |cos(2\pi f t)| \\
X_2(t) &= c_1 X_2(t-\tau_1) + c_2 X_1(t-\tau_2) + \eta_1(t) \\
X_3(t) &= c_3 X_1(t-\tau_3) * X_2(t-\tau_4) + \eta_2(t) \\
X_4(t) &= c_4 X_3(t-\tau_5) * \beta^{\frac{X_2(t-\tau_6)-Q}{10}} + \eta_3(t)
\end{aligned}$$

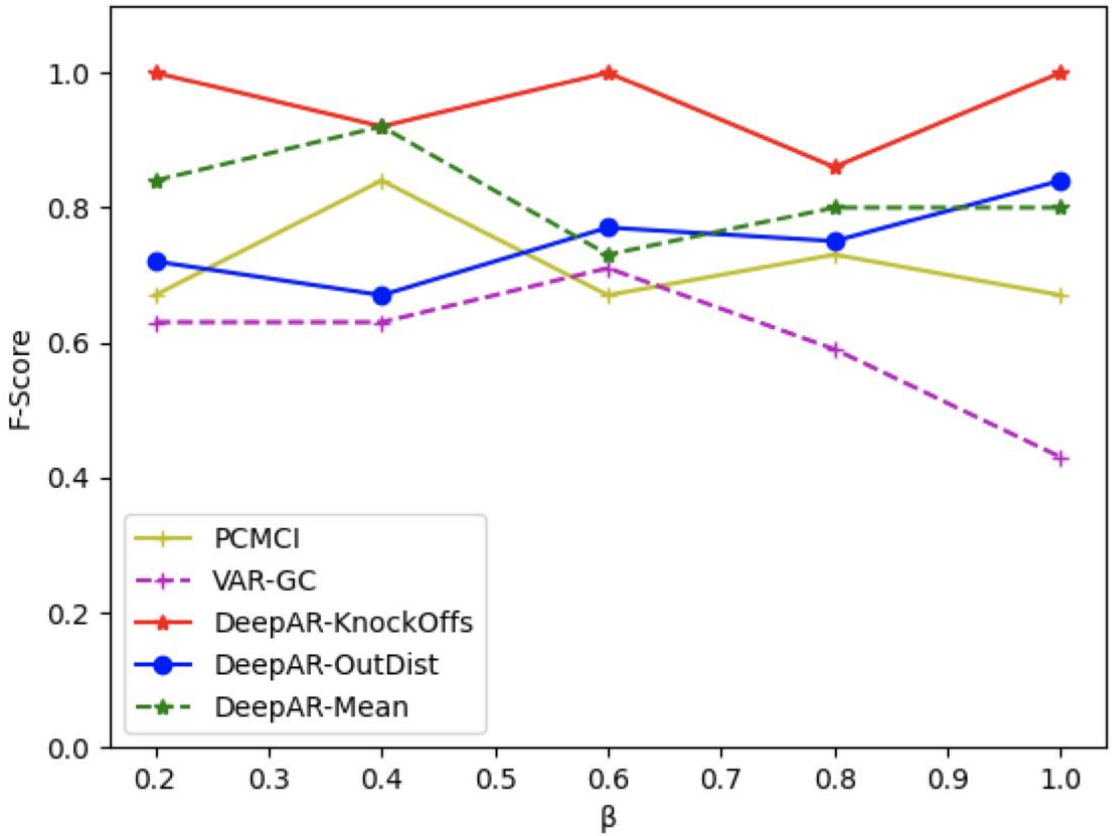
c = [0.95, 0.80, 0.50, 0.75]

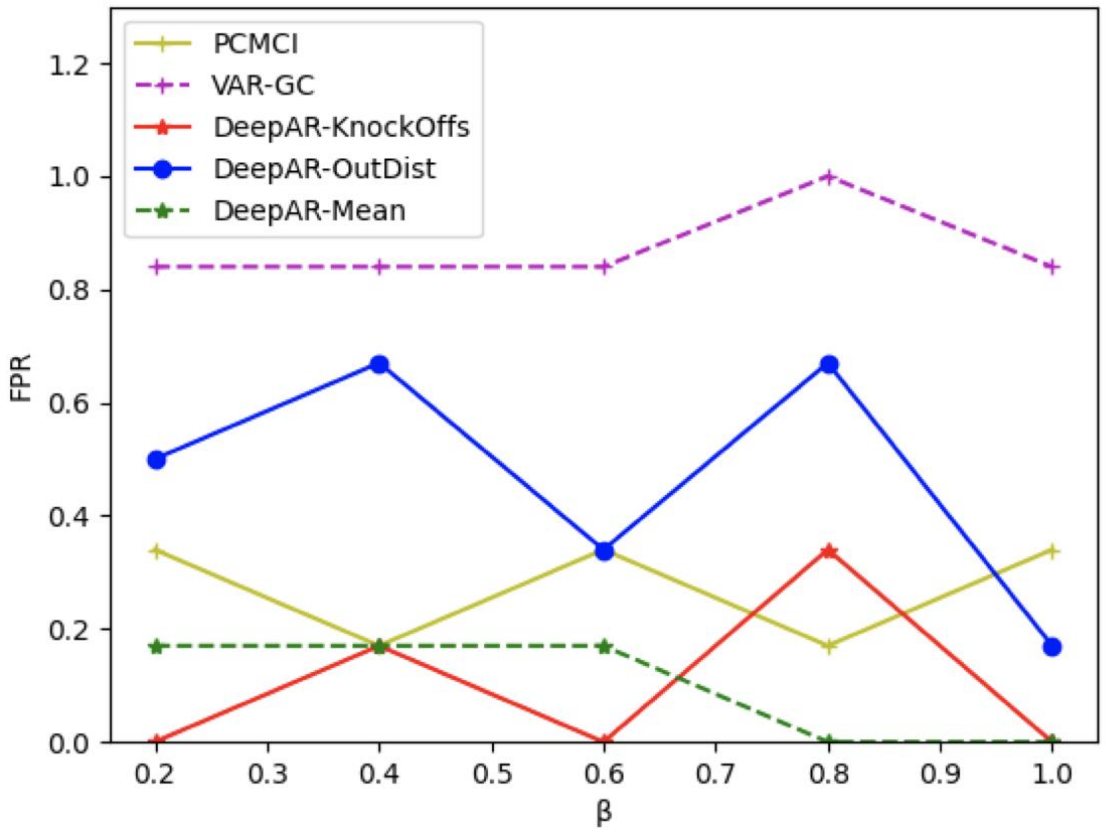
Q = 10

f = 150

0 <= t <= 3000

Эксперименты





Эксперименты

Реальные данные

Causal links	Expected	VAR-GC	PCMCI	DeepAR-Knockoffs
$K_t \to D_t$	Yes	Yes	Yes	Yes
$K_t \to L_t$	No	Yes	Yes	No
$D_t \to K_t$	No	Yes	Yes	No
$D_t \to L_t$	No	Yes	No	No
$L_t \to K_t$	No	Yes	Yes	No
$L_t \to D_t$	No	No	No	Yes

Небольшая реклама

ETNA Time Series Library

Многофункциональный удобный инструмент для работы с временными рядами

Небольшая реклама

- Большой зоопарк моделей, ансамбли
- Куча препроцессингов и расчетов признаков
- Много EDA
- Интеграция с Weights and Biases
- Работа с доп данными
- Возможность внедрять свои модели и трансформы
- Кластеризация рядов
- Поиск и обработка аномалий
- Работа с несколькими сегментами
- Визуализация всего что только можно
- Отбор признаков
- Подбор гиперпараметров
- Causal inference

Весь пайплайн

```
original_df = pd.read_csv("data/example_dataset.csv")
df = TSDataset.to_dataset(original_df)
ts = TSDataset(df, freq='D')

log = LogTransform(in_column="target")
trend = LinearTrendTransform(in_column="target")
seg = SegmentEncoderTransform()
lags = LagTransform(in_column="target", lags=list(range(30, 96, 1)))
d_flags = DateFlagsTransform(day_number_in_week=True,
                             day_number_in_month=True,
                             week_number_in_month=True,
                             week_number_in_year=True,
                             month_number_in_year=True,
                             year_number=True,
                             special_days_in_week=[5, 6])

train_ts, test_ts = ts.train_test_split(train_start='2019-01-01',
                                        train_end='2019-11-30',
                                        test_start='2019-12-01',
                                        test_end='2019-12-31')
train_ts.fit_transform([log, trend, lags, d_flags, seg])

model = CatBoostModelMultiSegment()
model.fit(train_ts)
future_ts = train_ts.make_future(HORIZON)
forecast_ts = model.forecast(future_ts)
```

ETNA

https://t.me/etna_support

https://github.com/tinkoff-ai/etna-ts

https://etna.tinkoff.ru