Structural time series grammar over variable blocks

David Rushing Dewhurst

Consider structural time series models that decompose additively. How can we extend these models to make them more expressive while still maintaining interpretability?

\[ y(t) = \varepsilon(t; \sigma) + \sum_{k} f_k(t; \theta_k) \]

\[ y(t) = \varepsilon(t; \sigma) + \sum_{k} f_k(t; \theta_k) + f_{k'}(t; \theta_{k'}) \]

We could add another “building block” term...

...or we could replace static parameters with further time varying components.

\[ \varepsilon(t; \sigma) \sim \text{Normal}(0, \sigma^2) \]

Implementation

Modeling

Small library of generative blocks that can be combined into valid sentences of the language generated by grammar \( G \) (with or without changepoint operator – with changepoint operator these are not causal models)

```
log_vol_1 = sts.AR1(t1=t1, ...)
log_vol_2 = sts.GlobalTrend(t1=t1, ...).cos()
vol = sts.changepoint(log_vol_1.exp(), log_vol_2.exp(), frac=0.6)
price = sts.RandomWalk(t1=t1, loc=0.0, scale=vol, ...).exp()
```

E.g., stochastic volatility model with changepoint + nontrivial latent structure

Implementation + W.I.P

Non-Markov DGP expressed in single block (same as simple first order Markov model)

Objects are stochastic -- can sample from whole STS or from component parts

Explicitly model decomposition is immediately interpretable (compare with GP kernel grammar)

Next steps: implement DSL + compiler to facilitate a) easier model expression and b) model search algorithms (searching for optimal string in language generated by grammar \( G \) subject to some constraints)

```
with stsb2.effects.ProposalEffect(trend):
    trend.parameter_update(a=posterior[trend]['a'],
        b=posterior[trend]['b'])
    trend_posterior = trend()
with stsb2.effects.ForecastEffect(...):
    trend_forecast = trend()
```

Inference

W.I.P. (only proof of concept LF rejection sampling), includes proposal, intervention, and forecast effect handlers for converting \( \text{sample(...)} \) statements into proposal and forecast distributions

Library: https://gitlab.com/daviddewhurst/stsb2; Documentation: https://davidrushingdewhurst.com/stsb2/docs/