THE UNIVERSITY OF BRITISH COLUMBIA

College of Graduate Studies

HMM Volatility Forecasting with Uncertainty-Aware Covariates

Duncan Cameron-Steinke

April 15 2026

Timeseries intro

Imagine you are a risk analyst. What's the probability of continued growth?

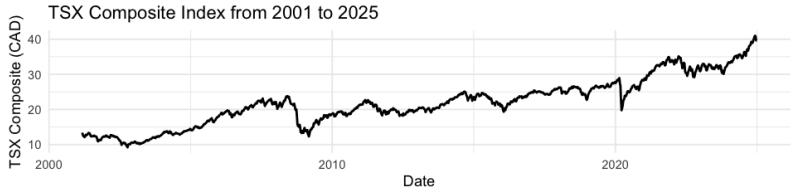

Figure: Friday closing price of TSX Composite Index from 2001 to Jan 1st, 2025.

Goals

- Identify historical market patterns.
- Extrapolate future conditions.

Timeseries intro

- Identify historical market patterns, using a Hidden Markov Model.

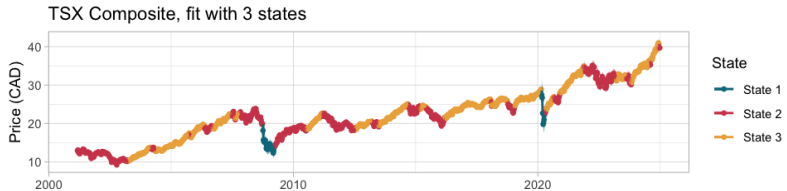

Figure: TSX fit with a 3-state Hidden Markov Model to identify distinct patterns.

- How to Extrapolate Future Conditions?

Outline

- ✓ Timeseries Intro
- ❑ Hidden Markov Models
- ❑ Deep learning Forecasts
- ❑ Evidential Deep learning
- ❑ Hybrid Model

Hidden Markov Models

Hidden Markov Model based on TSX composite.

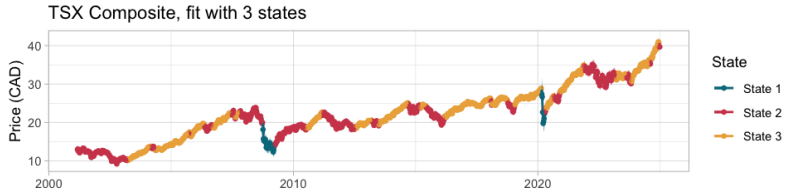

Figure: An HMM identifies lagent regimes in sequential data.

Hidden Markov Models

Hidden Markov Model based on TSX composite.

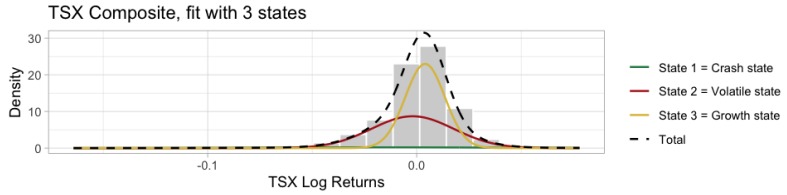

Figure: The states are analyzed and attributed to the corresponding market regime.

Hidden Markov Models

Assumptions

- The hidden state follows a Markov process

$$P(S_{n+1} \mid S_{1:n}) = P(S_{n+1} \mid S_n)$$

- Observations depend only on the current hidden state

$$P(Z_n \mid S_{1:n}, Z_{1:n-1}) = P(Z_n \mid S_n)$$

- The hidden state evolves based on the transition matrix

$$\begin{bmatrix} S_{1,n+1} \\ S_{2,n+1} \end{bmatrix} = \begin{bmatrix} p_{11}, & p_{12} \\ p_{21}, & p_{22} \end{bmatrix} \begin{bmatrix} S_{1,n} \\ S_{2,n} \end{bmatrix}$$

Hidden Markov Models

Hidden Markov Model based on TSX composite.

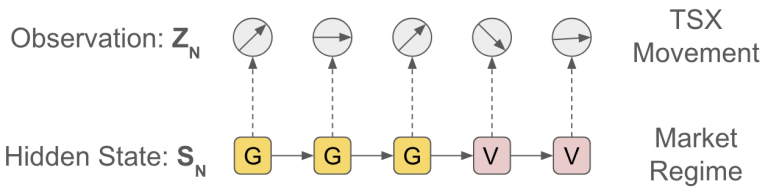

Figure: TSX Movement, indicated by arrow direction, is an observable random variable produced by the hidden market regime sequence which is either Growth (G) or Volatile (V).

Covariates

In many cases the observation sequence, and the hidden state sequence, are better explained with additional covariates.

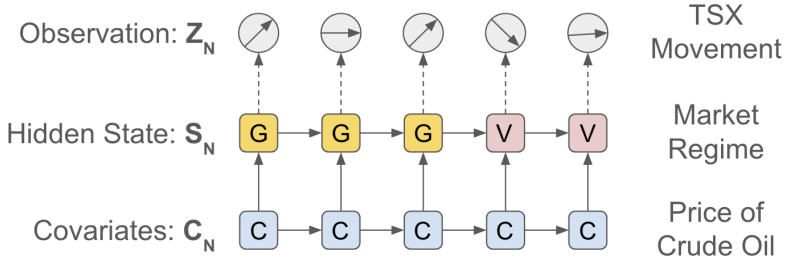

Figure: Price of Crude Oil acts as a covariate which can influence transition rates.

Covariates

- The observation sequence with covariates

$$Z_n \sim D(\theta_{\mathbf{p}})$$

$$\theta_{\mathbf{p}} = f(S_n, C_n)$$

- The hidden state transitions p_{ij} with covariates[1]

$$p_{ij} = \frac{g_{ij}(C_n)}{\sum_k^K g_{ik}(C_n)}$$

$$S_{n+1} = \mathbf{P}_n S_n$$

[1] g must be a positive real valued function.

HMM Forecasting

Forecasting extrapolates the hidden state based on the last known observation.

$$S_{N+1} = \mathbf{P}_N S_N$$

$$S_{N+2} = \mathbf{P}_{N+1} S_{N+1} = \mathbf{P}_{N+1} \mathbf{P}_N S_N$$

$$\dots$$

$$S_{N+T} = \mathbf{P}_{N+T-1} \mathbf{P}_{N+T} \dots \mathbf{P}_N S_N$$

The forecasted observation distribution is

$$Z_{N+T} \sim D(f(S_{N+T}, C_{N+T}))$$

HMM Forecasting

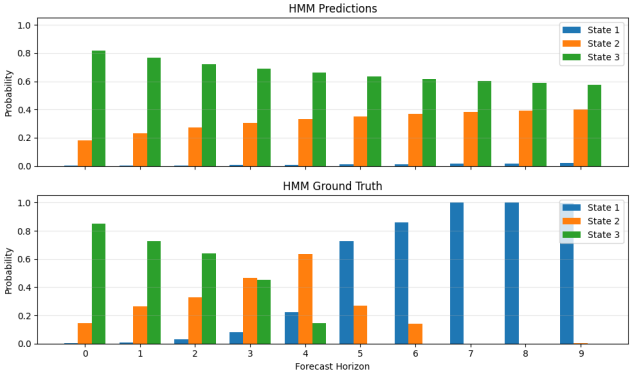

Figure: HMM Forecasts vs Ground Truth Labels, show that the HMM forecasts are accurate for short time horizons, but can not adapt to suden changes.

HMM Forecasting

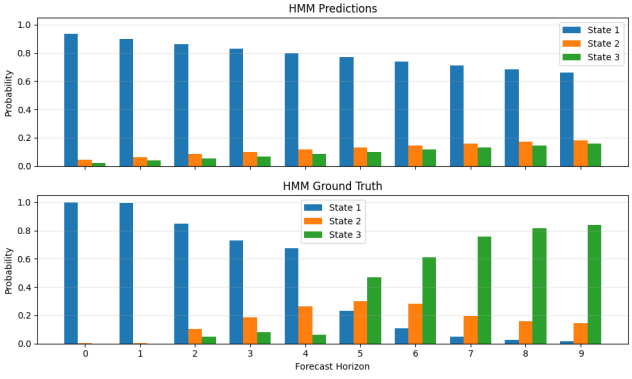

Figure: HMM Forecasts vs Ground Truth Labels.

HMM Forecasting

Forecasting presents three key problems.

1. In non-covariate models the forecasts are infelxible and revert to stationary distribution (long term average).

$$\bar{S} = \mathbf{P}\bar{S}$$

2. If Covariates are included then they must be known (or forecasted) into future periods, presenting new problems.
3. Covariate relationships must be modeled parametrically.

HMM Forecasting

Goal:

- Apply influence of covariates to future periods.
- Avoid extrapolating each covariate independently.
- Allow for non linear relationships between covariates.

Deep Learning Approach

Covariates with Train, Validation and Test splits

Covariates along with Training, Validation and Testing Splits

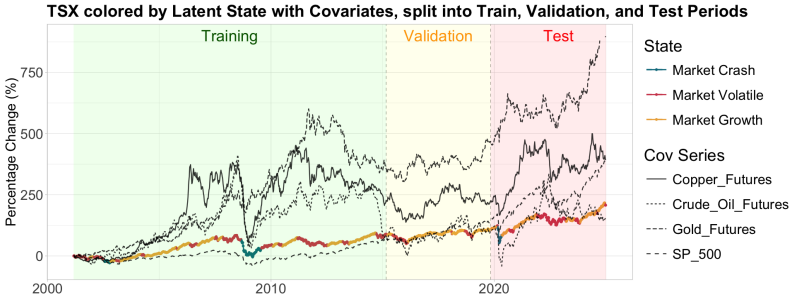

Figure: The HMM is used to obtain ground truth prediction labels, Validation period is used for tuning hyper-parameters, and Testing period provides an out of sample evaluation.

Deep Learning

1. Covariates with shape $[N_{Covariates}, T_{lookback}]$.
2. A Recurrent Neural Network (RNN) model encodes temporal features.
3. Interconnected dense layers learn complex non linear relationships between historical covariates and future hidden states.
4. Output layer returns the class probabilities.

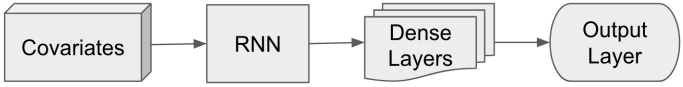

Figure: Deep Learning model which includes covariate inter-relationships.

HMM vs Deep Learning (RNN)

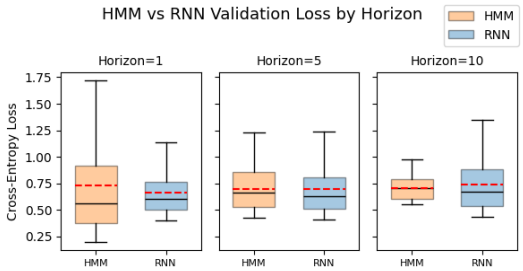

Figure: While RNN does slightly better for short time horizons, it is worse or equal to HMM for longer time horizons.

We want a method that selects the better model for each forecast.

Evidential Deep Learning

Generic Prediction

Softmax prediction output:

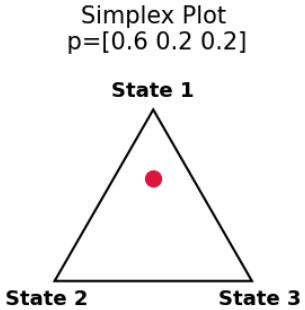

Figure: Sample distribution over labels for a generic classification output.

Evidence Based Prediction

Dirichlet distribution:

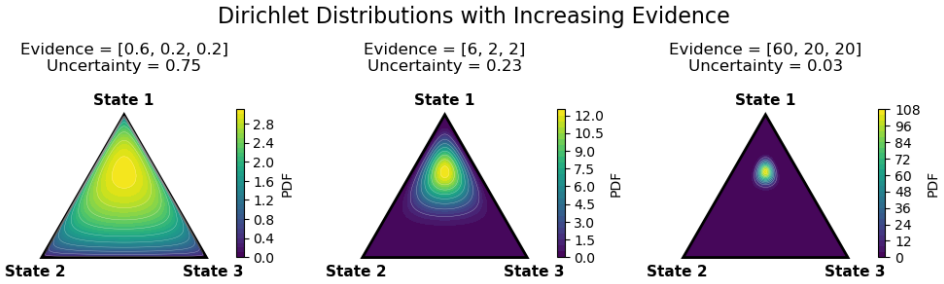

Figure: As evidence for each class increases, the uncertainty of the Dirichlet distribution decreases.

Evidence Based Prediction

The model generates and evidence scores via ReLu output[2].
Dirichlet parameters are

$$\alpha_i = e_i + 1$$

Confidence parameter:

$$S = \sum_{i=1}^{K} \alpha_i$$

Uncertainty:

$$u = \frac{K}{S}$$

[2]Sensoy 2018

Uncertainty types

Epistemic Uncertainty: When the model has insufficient information to make a confident prediction.

- Generally dominates in out-of-distribution examples.
  - Example: Covid-19 was a novel environment with high epistemic uncertainty.

$\rightarrow$ Predictions should have low confidence.

Aleatoric Uncertainty: Randomness inherent to the data itself.

- Should lead to more probability mass spread across the hidden states.
  - Example: Random week-to-week swings of an asset price.

$\rightarrow$ Predictions can have high confidence.

Evidential Deep Learning

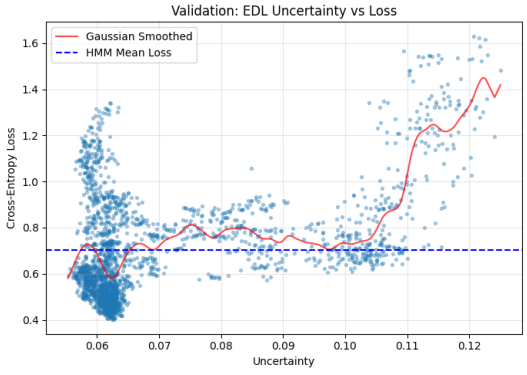

Figure: Evidential Deep Learning Cross-Entropy Loss vs Uncertainty, shows a positive corelation.

Evidential Deep Learning

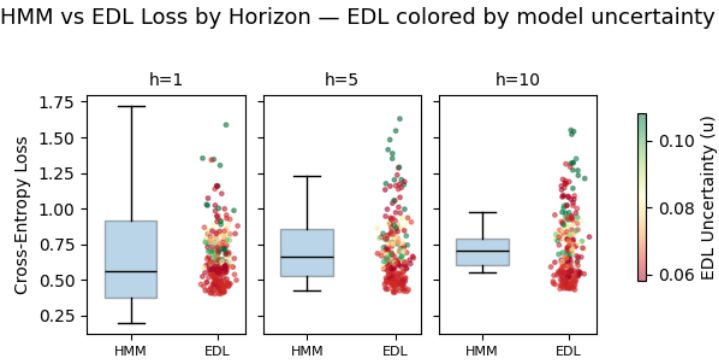

Figure: Low uncertainty predictions tend to produce lower loss across time horizons.

Hybrid Model

Hybrid HMM + EDL

Hybrid Goal

- Leverage EDL, when EDL uncertainty is low, use HMM otherwise.

$$p_{ij}^{*} = EDL_j \cdot \beta(u) + HMM_{ij}(1 - \beta(u))$$

Where

- p_{ij}^{*} is the hybrid transition probability from state i to j
- EDL_j is the EDL prediction for state j
- $\beta(u) = \beta_0 - u\beta_1$ is a tunable mixing parameter with hyper-parameters β_0, β_1 that selects when to use EDL vs HMM as a function of uncertainty u.
- HMM_{ij} is the native HMM transition matrix.

Hybrid HMM + EDL

Tuning the mixing parameter $\beta(u)$.

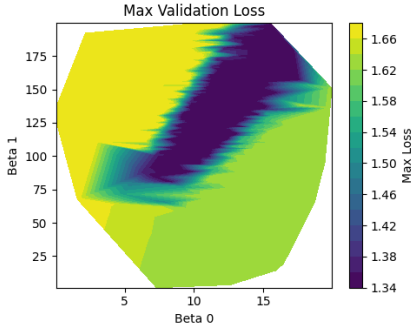

Figure: Max validation loss is optimized for mixing parameter values $\beta_0 = 11, \beta_1 = 130$

Results

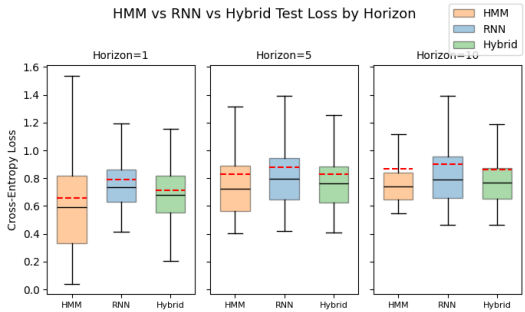

Figure: Box plot for hybrid model show slightly improved results but not significantly different.

Results

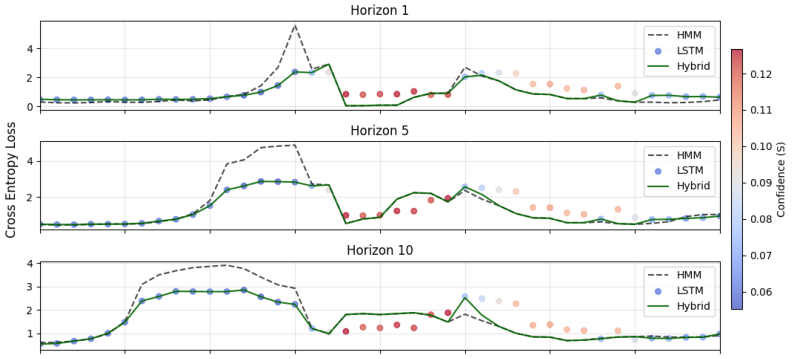

Figure: Time Series comparisons for HMM, RNN, and Hybrid models. Showing that the Hybrid model benefits from EDL when confidence is high while tracking HMM forecasts otherwise.

Conclusion

- HMMs are essential for historical time series, but not well suited for forecasting.
- Hybrid HMM + EDL benefits from deep learning when confidence is high.
- The Hybrid model can be interpreted like an HMM, while still benefiting from covariate information.

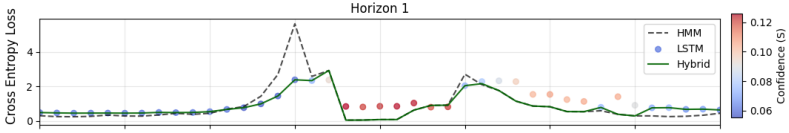

Figure: Hybrid Model incorporates EDL only when confidence is high.

Questions?

References

hmmTMB: Hidden Markov Models with Flexible Covariate Effects in R

Author: Théo Michelot
Journal: Journal of Statistical Software
Year: 2025
Volume: 114(5)

Evidential Deep Learning to Quantify Classification Uncertainty

Author: Murat Sensoy, Lance Kaplan, Melih Kandemir
Proceedings: 32nd Conference on Neural Information Processing Systems.
Year: 2018

EDL Loss function

Sensoy proposes the following categorical cross entropy loss function with a dirichlet prior:

$$\mathcal{L}_i(\Theta) = \int \left[\sum_{j=1}^{K} -y_{ij} \log(p_{ij}) \right] \frac{1}{B(\boldsymbol{\alpha}_i)} \prod_{j=1}^{K} p_{ij}^{\alpha_{ij}-1} \, d\mathbf{p}_i = \sum_{j=1}^{K} y_{ij} \big(\psi(S_i) - \psi(\alpha_{ij}) \big),$$

- $\sum_{j=1}^{K} -y_{ij} \log(p_{ij})$ is the categorical cross entropy.
- $\frac{1}{B(\boldsymbol{\alpha}_i)} \prod_{j=1}^{K} p_{ij}^{\alpha_{ij}-1}$ is the dirichlet PDF, serving as conjugate prior.
- $\psi(\cdot)$ is the digamma function.

EDL Loss function + Regularization

I experimentally found that adding a regularization term λS_i which adds a penalty for high confidence (aka low uncertainty) improved hybrid model performance by forcing the model to default to having high uncertainty unless it had strong evidence to select otherwise.

$$\mathcal{L}_i(\Theta) = \sum_{j=1}^{K} y_{ij}\big(\psi(S_i) - \psi(\alpha_{ij})\big) + \lambda S_i$$