Latent Representation in Financial Markets

Tianle Chen, Sebestian Jaimungal (University of Toronto),
Abe Chan (TMX Group)

Table of contents

Introduction

Objective

- We explore data-driven static and dynamic representations of trading conditions to visualize and understand data-driven relations between *orders, book and trades*.
- To achieve this, we reconstruct *static features and temporal dynamics* using neural networks and observe their internal state.

Applications

- Interpretable Forecasts
- Anomaly Detection
- Generating Synthetic Features

Static Representation

Reconstruction

- We assume that a large number covariates is sufficient to represent market conditions.
- For each time step, we reconstruct the features by representing them on a latent space using a *decoder* and an *encoder*.
- On a latent space, we can then cluster the observations and explore behaviours captured by the clusters.

Variational Auto-Encoder

We represent the features (X) using a pair of neural networks: an encoder and a decoder, which are non-linear functions. Z is the latent state.

- $g_{\text{decoder}} : Z \to X$
- $f_{\text{encoder}} : X \to$ (mean, log-variance), $Z|X \sim \mathcal{N}$(mean, log-var).

$$
\begin{tikzcd}
Z|X \arrow[rr, bend right, "g_{\text{decoder}}"'] & & X \arrow[ll, bend right, "f_{\text{encoder}}"']
\end{tikzcd}
$$

Figure 1: Variational Auto-Encoder

Note that `outputs` $\sim \mathcal{N}$(`z_mean, z_log_var`).

```python
## ----------------------------------------------------------------------------
## instantiate VAE model
## ----------------------------------------------------------------------------
outputs = decoder(encoder(inputs)[2])
vae = Model(inputs, outputs, name='vae')

# VAE loss = mse_loss or xent_loss + kl_loss
# losses defined in terms of [inputs, outputs], which point to keras layers
reconstruction_loss = ncov*mse(inputs, outputs)
kl_loss = 1 + z_log_var - K.square(z_mean) - K.exp(z_log_var)
kl_loss = K.sum(kl_loss, axis=-1)
kl_loss *= -0.5

vae_loss = K.mean(reconstruction_loss + kl_loss)
```

Figure 2: VAE Loss[1]

[1] https://github.com/tianle91/forecastor/blob/master/rep_vae.py

Clustering with GMM

- We segment points on the latent space into clusters to visualize distinct behaviour.
- Given g clusters, $\{\mu_i, \Sigma_i\}_{i=1}^{g}$ defines g Gaussians and their respective mixture weights.
- Parameters given g are selected by maximizing likelihood.
- The optimal GMM and associated number of components g is selected by minimizing cross-validation loss.

Data

- We ran the model on `TSX:TD` for full-day minute-level features during the period `2018-04-01` to `2018-05-01`.
- This yields 22 days of data with 390 time-steps per day.
- There are 149 features collected from tables `cbbo, orderbook_tsx, trades`[2].

[2] `https://raw.githubusercontent.com/tianle91/forecastor/master/covnames.txt`

Data Visualization

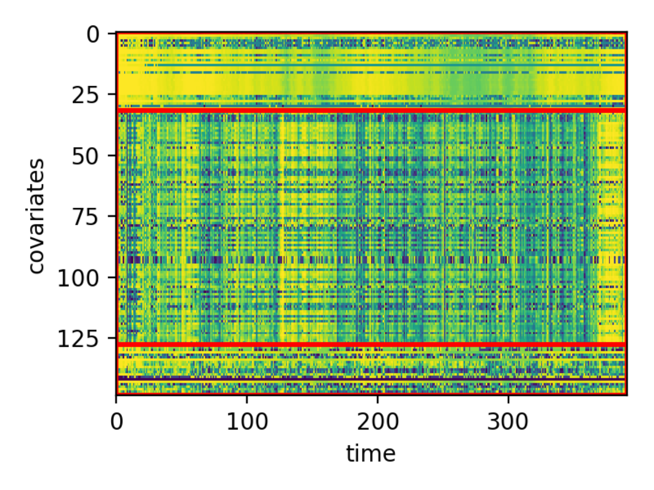


Figure 3: Covariates for `TSX:TD` on `2018-04-01`

VAE Structure: Encoder

The input layer is 149 dimensions while the intermediate layer is 64. The latent representation is 32 dimensions.

$$\text{input } (X \in \mathbb{R}^{149}) \longrightarrow \text{interm } (64) \longrightarrow \text{mean}(32), \text{log-var}(32)$$

Figure 4: Architecture of encoder. We will use the mean vector as latent representation.

Clustering on Mean

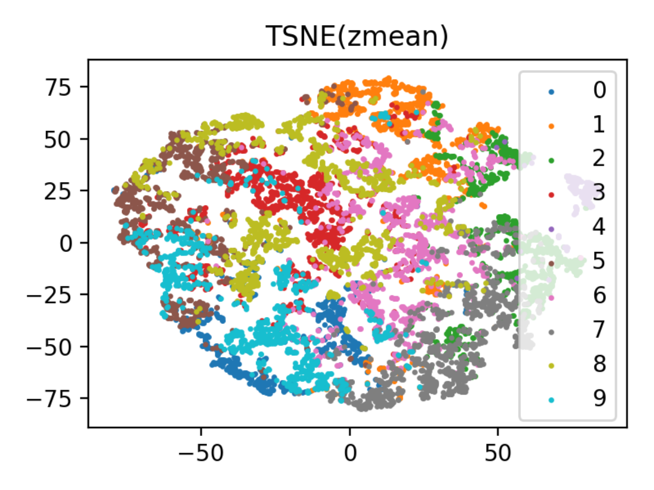


Figure 5: TSNE with clusters colored. Number of components identified here (9) is chosen from GMM+CV in $\mathbb{R}^{32}$ over 1-32 components.

Clusters and Time of Day

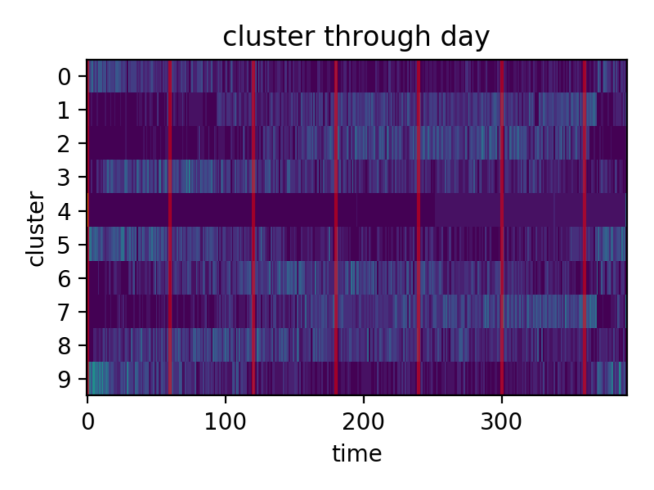
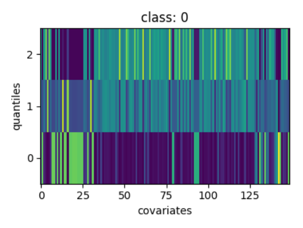


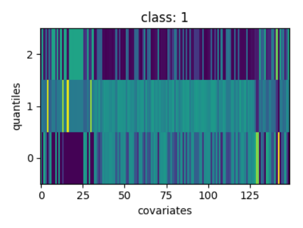
Figure 6: Clustering by Time of Day. Hourly intervals are denoted by vertical lines. Morning and afternoon behaviour are captured by these clusters.

Clusters and Distribution of Covariates

We visualize how observations for each cluster are different by counting the number of observations which fall in every quantile.

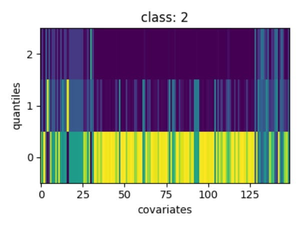


(a) Class 0*

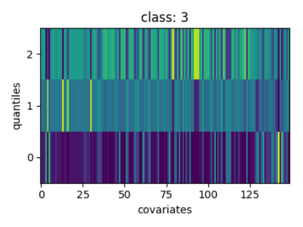


(b) Class 1

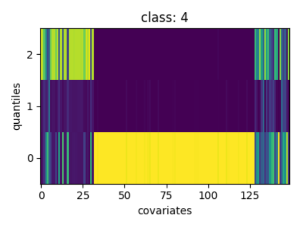
Clusters and Distribution of Covariates



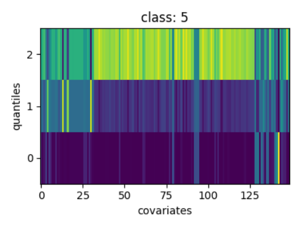
(a) Class 2*



(b) Class 3*

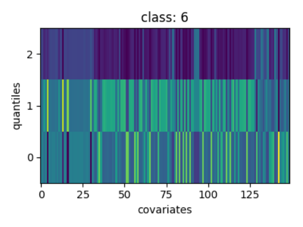


(c) Class 4*

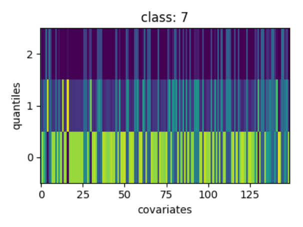


(d) Class 5

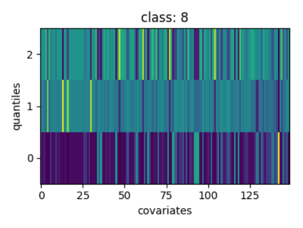
Clusters and Distribution of Covariates



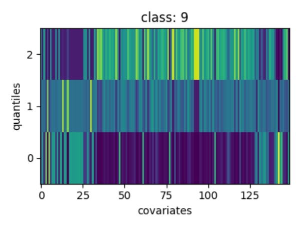
(a) Class 6



(b) Class 7

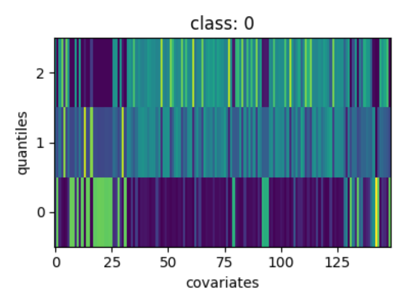


(c) Class 8



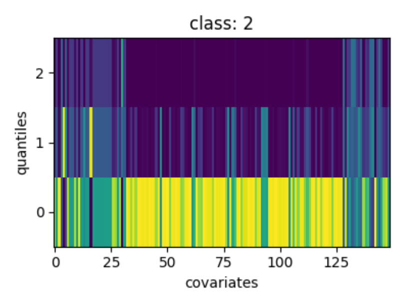
(d) Class 9

Notable Clusters: Class 0



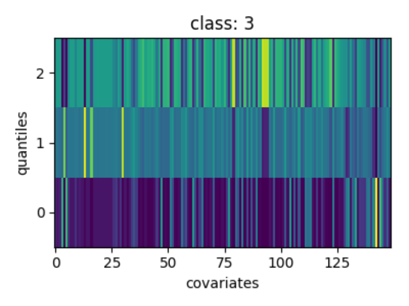
Quantile	Covariate
Medium	max spread
Medium	min spread
High	volume of sell-side book-changes at touch
High	volume of new sell-side orders at touch
Low	min price_diff2

Notable Clusters: Class 2



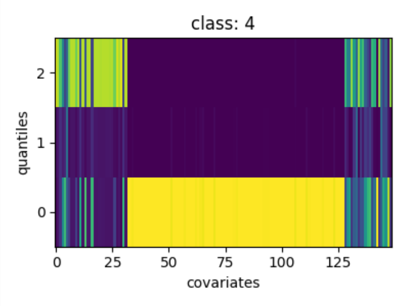
Quantile	Covariate
Low	number of all book-changes at touch (10-ticks)
Low	number of new orders at touch (10-ticks)
Low	number of new orders at touch (5-ticks)
Low	number of all book-changes at touch (5-ticks)
Low	min price_diff2

Notable Clusters: Class 3



Quantile	Covariate
Medium	max spread
Medium	min spread
High	number of buy-side book-changes at touch
High	number of new buy-side orders at touch
Low	min price_diff2

Notable Clusters: Class 4



Quantile	Covariate
High	[book covariates]
Low	[book changes covariates]
-	-
-	-
-	-

Temporal Dynamics

Prediction

- To capture temporal dynamics, we predict the next-step feature of interest using a recurrent neural network (RNN).
- On an intermediate state of the RNN, we can then cluster the observations and explore behaviours captured by the clusters.

RNNs make sequential predictions (Y_t) by continually updating a memory state (Z_t) using covariates (X_t).

$$(X_t, Z_{t-1}) \xrightarrow{\quad f \quad} (Z_t, Y_t)$$

Figure 10: Recurrent Neural Network

Applying this over a sequence of $\{X_t\}_{t=0}^{T}$ returns a sequence of predictions $\{Y_t\}_{t=0}^{T}$ as well as memory states.

Data

- We ran the model on `TSX:TD` for full-day minute-level features during the period `2018-04-01` to `2018-05-01`.
- We chose a sliding window of 1 hour with 1-minute intervals.
- Over a period of 1 month and full-day data, we obtain a total of 7238 sequences. The last day (329 sequences) is used for validation.

Response Variable

- We predict next-minute `quadratic variation`.
- Response is discretized in empirical quantiles so this is a multi-class prediction problem. Splitting into 3 quantiles give intuitive labels: Low, Medium, High.
- Response at current state is included among covariates.

RNN Structure

- The input layer is 149+3=152 dimensions while the recurrent layer is 32 dimensions.
- Recurrent map is a Gated Recurrent Unit [3].
- Activated output is pmf over 3 classes (2+1 dimensions).

$$\text{input } (X_{t-1} \in \mathbb{R}^{152}) \longrightarrow \text{GRU}(32) \longrightarrow \hat{Y}_t \in \mathbb{R}_+^{2+1}$$

Figure 11: Predicting next-step response with RNN.

[3] `https://keras.io/layers/recurrent/`

Prediction Baselines

- Previous value: $\hat{Y}_t = Y_{t-1}$
- Constant value: $\hat{Y}_t = c$
- Markov chain: $\hat{Y}_t = P_{\text{training}}(Y_t|Y_{t-1})$

	$P_{\text{training}}(Y_t\|Y_{t-1})$		
Y_{t-1}	Low	Med	High
Low	0.68	0.26	0.04
Med	0.28	0.48	0.22
High	0.04	0.27	0.68

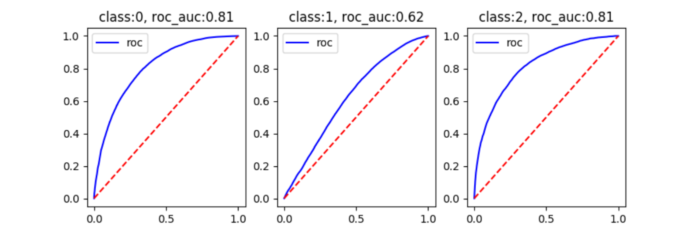
Figure 12: Markov Transitions in training set

- Logistic Regression: $\hat{Y}_t = \beta^\top W_{t-1}$ where $W_{t-1} = [1, X_{t-1}, Y_{t-1}]$.

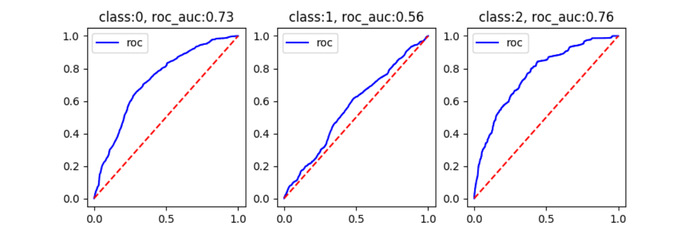
Prediction Evaluation

- Receiver Operator Characteristics (ROC) curve plots false positive rate against true positive rate by varying threshold values of a predictor.
- Area Under Curve (ROC-AUC) represents the quality of a binary predictor.

Prediction Evaluation for RNN



(a) Training



(b) Testing

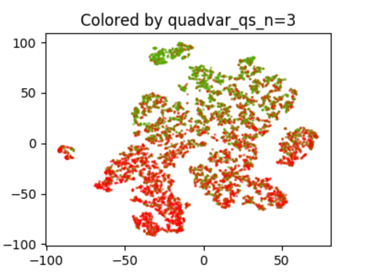
Evaluation Comparison

Model	Training				Testing			
	Low	Med	High	Avg	Low	Med	High	Avg
Constants	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.59
Previous	0.67	0.55	0.69	0.63	0.64	0.55	0.64	0.61
Markov	0.72	0.55	0.73	0.66	0.67	0.52	0.67	0.62
Logistic	0.79	0.61	0.80	0.73	0.72	0.56	0.71	0.66
RNN	0.81	0.62	0.81	0.74	0.73	0.56	0.76	0.68

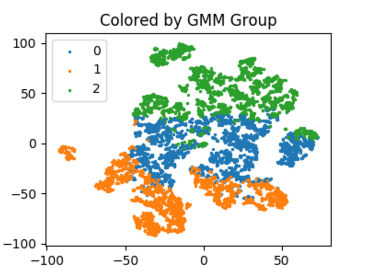
Figure 14: ROC-AUC performance comparisons.

RNN Latent Representation

We consider the pre-activation layer at the last time-step as representative of the market conditions during the 60-minute period that relates to quadratic variation.



(a) Colored by response.

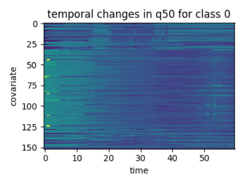


(b) Colored by fitted GMM ($g = 3$).

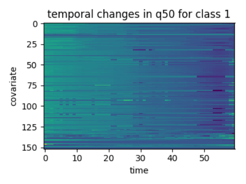
Figure 15: Clustering in the pre-activation layer segments sequences that predict next time-step quadratic variation. (Green is High)

Temporal Changes in Clustered Observations

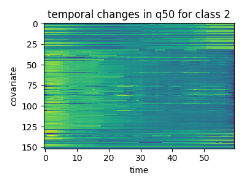
- We find sequences for each class and compute the median values for each covariate at each time step.
- Separate normalizations are done for each class so that in-cluster temporal differences are emphasized within the 60-minute time window.



(a) Class 0: High

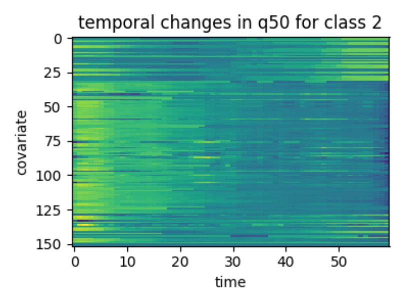


(b) Class 1: Medium



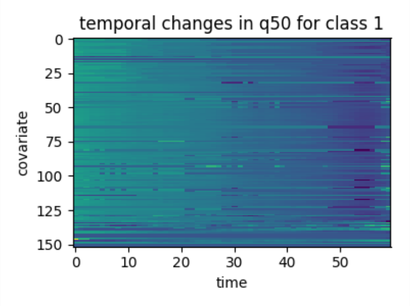
(c) Class 2: Low

Temporal Changes: Cluster 2 (Low)



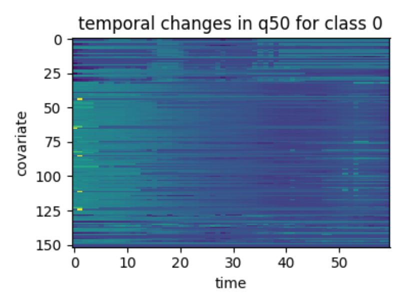
Minute	Covariate
1	vol-weighted-price_diff2
1	mean logreturn
1	mean price_diff2
2	mean logreturn
3	mean logreturn

Temporal Changes: Cluster 1 (Medium)



Minute	Covariate
0	sum price_diff2
59	volume of executed orders at touch (5-ticks)
59	volume of executed orders at touch (50-ticks)
59	volume of executed orders at touch (10-ticks)
59	volume of executed orders

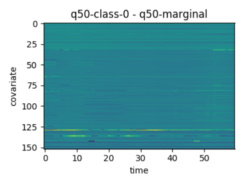
Temporal Changes: Cluster 0 (High)



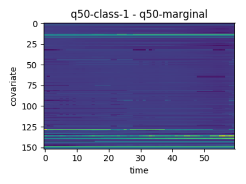
Minute	Covariate
0	number of executed sell-side orders at touch (10-ticks)
1	number of executed buy-side orders
1	number of executed buy-side orders at touch (10-ticks)
1	number of executed buy-side orders at touch (50-ticks)
1	number of executed buy-side orders at touch (5-ticks)

Between-cluster comparisons

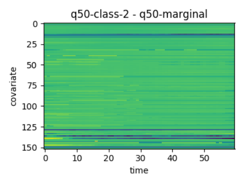
- We compute the difference between median values for each covariate and each time step and the population-wide median values.
- Normalization is done for each time step and covariate, so differences between clusters are observable.



(a) Class 0: High

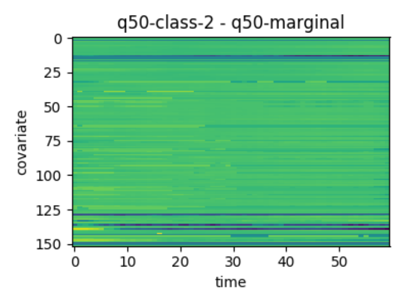


(b) Class 1: Medium



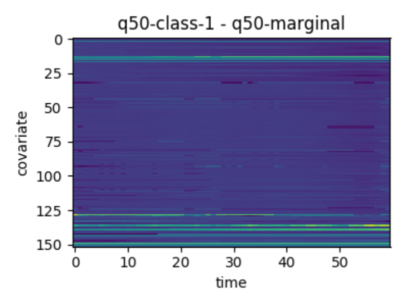
(c) Class 2: Low

Between-cluster: Cluster 2 (Low)



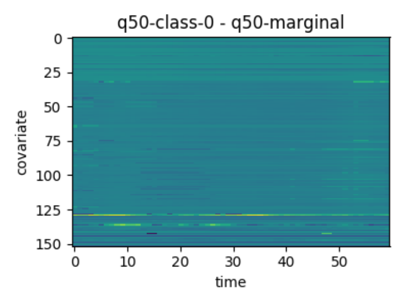
Minute	Covariate
0	max log-return
1	max log-return
2	max log-return
3	max log-return
16	max quantity

Between-cluster: Cluster 1 (Medium)



Minute	Covariate
0	min quantity
55	mean touch-weighted-price logreturn
57	mean touch-weighted-price logreturn
58	mean touch-weighted-price logreturn
59	mean touch-weighted-price logreturn

Between-cluster: Cluster 0 (High)



Minute	Covariate
8	vol-weighted logreturn
9	vol-weighted logreturn
33	vol-weighted logreturn
34	vol-weighted logreturn
35	vol-weighted logreturn

Future Work

Future Work

- Reducing number of covariates through feature selection.
- Joint modeling between various correlated stocks.
- Interpretability-focused clustering.
- Robust latent representations.