GLOBAL VOLATILITY SUMMIT

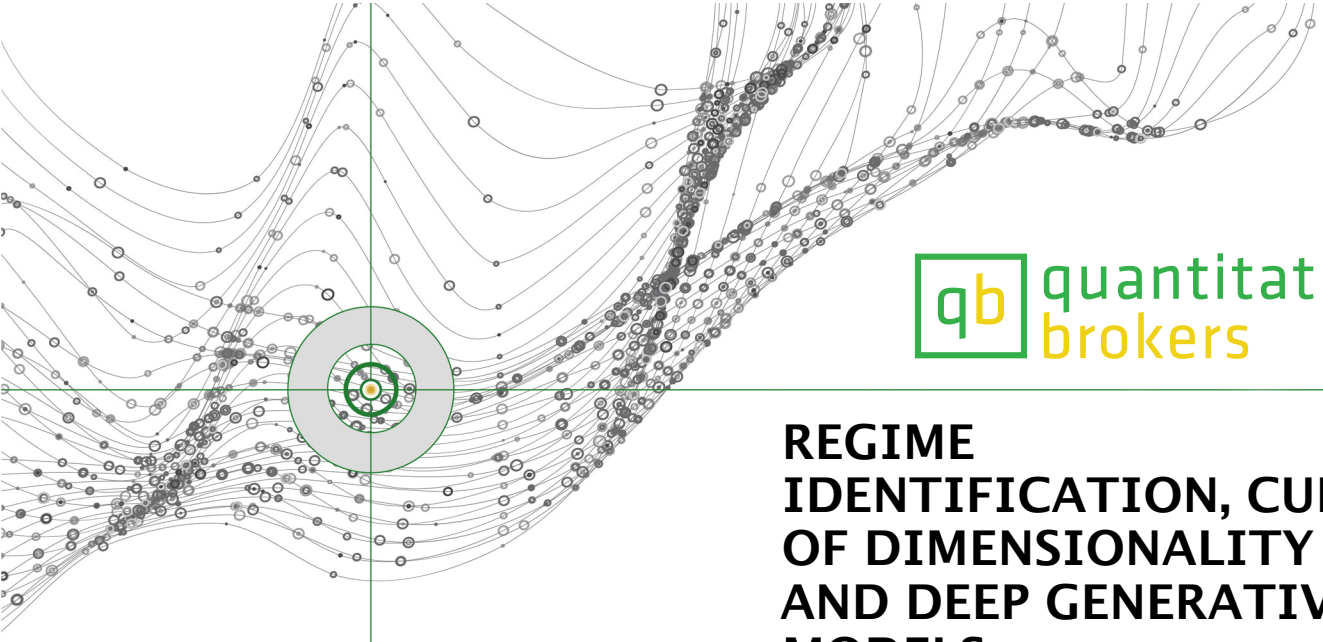
August 2021 Newsletter

Dear Investor,

The Global Volatility Summit ("GVS") brings together volatility and tail hedge managers, institutional investors, thought-provoking speakers, and other industry experts to discuss the volatility markets and the roles volatility strategies can play in institutional investment portfolios. The GVS aims to keep investors updated on the volatility markets throughout the year, and educated on innovations within the space.

Quantitative Brokers has provided the latest piece in the GVS newsletter series.

Cheers,
Global Volatility Summit



quantitative brokers

REGIME IDENTIFICATION, CURSE OF DIMENSIONALITY AND DEEP GENERATIVE MODELS

AMAN SAWHNEY
FEBRUARY 23, 2021

Executive Summary:

- High-frequency data facilitates clustering different products and identifying regimes for execution purposes.
- Previous QB white papers have addressed regimes in execution performance using microstructure variables.
- The problem with high-frequency data is the curse of dimensionality.
- Secondly, high-frequency data is noisy, and the relationship between the input variables such as quote size, volatility, and liquidity, etc., is not linear.
- Simple clustering techniques are inadequate in dealing with the noise in the input features.
- This paper refers to an unsupervised Neural Network method to cluster instruments.
- Forthcoming papers will use this clustering technique for cost estimation of products across different regimes and even exchanges.

INTRODUCTION

Market movements are motivated by an uncountable number of factors. To model these factors, one must work in extremely noisy high dimensional spaces. However, most tractable methods of analysis require low dimensionality. As a result, feature identification and feature engineering largely determine the effectiveness of most financial models. Unfortunately, due to the excess prevalence of non-linearities and non-experimental nature, feature engineering for financial datasets is a laborious and challenging task. Fortunately, deep neural methods do not suffer from this problem. Deep neural methods can automatically discover non-linear feature maps and produce low-dimensional spaces for learning tasks [6]. These features can be used directly for unsupervised learning tasks or used as inputs for supervised learning tasks [6].

This paper explores the motivation for dimensionality reduction using autoencoders and variational autoencoders, provides a brief overview of variational autoencoders and autoencoders, and introduces a novel application of variational autoencoders. Namely, we explore the potential of using variational autoencoders for performing execution regime identification.

MOTIVATION FOR NON-LINEAR DIMENSIONAL REDUCTION

In our previous paper [7], we introduced the concept of multidimensional regimes. We showed that regimes could impact arrival price slippage. Our focus was on using different microstructure variables such as quote size, volatility, liquidity, etc. We used simple k-means clustering to determine different regimes in an unsupervised manner. Our multidimensional regime's output identifies the current regime and recommends a particular execution algorithm for that regime. However, there are several other microstructure variables with non-linear relationships. To capture these non-linear relationships and further understand the various execution performance regimes, we must utilize a new learning paradigm. This paper partly addresses these issues using an unsupervised neural network method of effectively combining the input features.

MODELS

AUTOENCODERS AND VARIATIONAL AUTOENCODERS

An Autoencoder is a learning network that attempts to reconstruct its input using a lower-dimensional "bottlenecked" space. The initial mapping to the lower-dimensional space is referred to as the encoder; the mapping from the lower-dimensional space to the reconstructed input is known as the decoder [6]. Applications of autoencoders range from pretraining deep neural networks to data noise reduction [1]. Figure 1 illustrates a simple autoencoder network. However, since traditional autoencoder networks focus only on reconstructing the original data, the latent dimensions are not ensured to be continuous. As a result, the latent space may not be useable for tasks that rely on continuity.

Variational Autoencoders (VAE) slightly modify the autoencoder paradigm. Instead of outputting an encoded representation directly, the encoder output parameters to a probability distribution. This distribution is then sampled, and the decoder attempts to rebuild the original input based on the sampled value. Within this framework, the latent variable is assumed to be continuous [5]. Hence, the resulting learned latent space is continuous. As a result, variational autoencoders can produce far more useful spaces for tasks that range from content generation to interpolation. However, variational autoencoders are less general than vanilla autoencoders as they enforce a known prior distribution.

VARIATIONAL AUTOENCODER WITH GAUSSIAN PRIOR (VAE)

A varational autoencoder assumes $p(x,z) = p(x|z)p(z)$ as the generative model for any given data point, x, and latent point, z. In its most simplest form, we assume a Gaussian prior for the latent variable z. Thus we have $z \sim \mathcal{N}(0,1)$ and since we have real valued data, $x|z \sim \mathcal{N}(\mu_z, \sigma_z)$ [5].

We wish to train our varational autoencoder to model $p(z|x)$(encoder) and $p(x|z)$(decoder). By Bayes rule we have $p(z|x) = \frac{p(x|z)p(z)}{p(x)}$. Further $p(x) = \int_z p(x|z)p(z)dz$. Unfortunately, this integral would require exponential time to compute for each point in our latent space. Hence, calculating $p(x)$ is intractable. We can resolve this by instead using a neural network to model another distribution, $q_\lambda(z|x)$ such that $q_\lambda(z|x)$ is close to $p(z|x)$. To measure the similarity between two distributions we can use the Kullback–Leibler Divergence [3]. Formally, we define $D_{KL}(P||Q) = \mathbf{E}_P[\log P(x) - \log Q(x)]$ [3]. Now we wish to find λ^* such that $q_{\lambda^*}(z|x) =$

$\min_\lambda D_{KL}(q(z|x)||p(z|x))$ This can be achieved by maximizing the evidence lower bound (ELBO) [5]. We define the ELBO as: $\mathbf{ELBO}(\lambda) = \mathbf{E}_{q_\lambda(z|x)}[\log p(x|z)] - D_{KL}(q_\lambda(z|x)||p(z))$

Hence, we can train our autoencoder using

$$l(\lambda) = -\mathbf{ELBO}(\lambda) \tag{1}$$

In the context of an autoencoder, $\mathbf{E}_{q_\lambda(z|x)}[\log p(x|z)]$ is the reconstruction error of our network. Thus we can regard a variational autoencoder as a variant of an autoencoder that regularizes the latent space to follow a known prior probability distribution.

VARIATIONAL AUTOENCODER WITH GAUSSIAN MIXTURE PRIOR (GMVAE)

To enforce a Gaussian mixture prior, we will assume the generative model, $p(x,z,c) = p(c)p(z|c)p(x|z,c)$. We assume $c \sim \mathbf{Cat}(\frac{1}{K})$, $z|c \sim \mathcal{N}(\mu_c, \sigma_c)$, and since our data is real valued we additional assume $x|z,c \sim \mathcal{N}(\mu_{z,c}, \sigma_{z,c})$. Again we wish to maximize the ELBO. Following [9], [2] we thus have

$$l(\lambda) = \sum_y q(y|x)(\log q(y|x) - \log p(y) + \log q(z|x,y) - \log p(z|y) - \log p(x|y,z)) \tag{2}$$

Note, $\sum_y q(y|x)(\log q(y|x) - \log p(y))$ enforces the prior distribution of the categorical variable, $\sum_y q(y|x)(\log q(z|x,y) - \log p(z|y))$ enforces the Gaussian prior of each mixture component, and $\sum_y q(y|x)(-\log p(x|y,z))$ is the negative log likelihood of our reconstructed data.

DISENTANGLING THE LATENT REPRESENTATION (β-VAE)

We can additionally view maximizing **ELBO** as a constrained optimization problem where we wish to maximize the reconstruction error of our data such that our latent embedding follows a known probability prior. More rigorously, we wish to $\max \mathbf{E}_{q_\lambda(z|x)}[\log p(x|z)]$ such that $D_{KL}(q_\lambda(z|x)||p(z)) < \epsilon$ for some small ϵ [4]. If we attempt to solve this using Lagrange multipliers, we must maximize $\mathbf{E}_{q_\lambda(z|x)}[\log p(x|z)] - \beta(D_{KL}(q_\lambda(z|x)||p(z)))$ where β is our Lagrangian. We may view β as a tunable hyperparameter which regularizes our latent space. If we increase β, our encoder must use our latent space more efficiently. Since disentangled representations are encoded more efficiently, increasing β forces the autoencoder to disentangle the latent space [4].

Thus we can modify (2) to promote disentanglement of the latent space. Hence we obtain

$$l(\lambda) = \sum_y q(y|x)(\beta * (\log q(y|x) - \log p(y) + \log q(z|x,y) - \log p(z|y)) - \log p(x|y,z)) \tag{3}$$

REGIME DETECTION METHOD

We posit that different clusters of market microstructure variables exhibit different execution performance. We suggest using a beta gaussian mixture variational autoencoder to cluster our high dimensional market microstructure dataset. For this research, we will consider the problem of identifying different execution performance regimes for E-mini S&P 500 futures.

We will make use of both market data and order data collected by Quantitative Brokers. Specifically, our market dataset consists of one-minute bined E-mini S&P 500 features

related to 50-minute, 20-minute, and 15-minute volatility, quote size, average executed order size, and amihud illiquidity from November 1, 2018, to December 28, 2020. In total, our feature space is 30-dimensional. Additionally, we use an isolation forest to remove any outliers from our market dataset and take each feature's cubic root to scale each feature robustly. Our order data consists of parent order features for orders executed by Quantitative Brokers on the Chicago Mercantile Exchange from November 1, 2020, to December 28, 2020, with an order size greater than two.

First, we will train a β-GMVAE to learn a lower-dimensional latent representation of market data from November 1, 2018, to November 1, 2020. Then we will use the posterior cluster assignments of our embedded latent variable as our unsupervised clusters. Finally, we will use our model to classify regimes for order data from November 1, 2020, to December 28, 2020, and analyze those orders' market microstructure. Figure 2 shows the regime identification model in more detail.

MODEL DECISIONS

We opted to encode into a two-dimensional latent space, use four mixture components, and $\beta = 3$. We noticed a fair bit of instability in our latent space. To address this, following previous research with generative adversarial networks, we opted to include a batch normalization layer after each dense layer and used a tuned optimizer [8].

RESULTS

We trained our final β-GMVAE on our market data from November 1, 2018, to November 1, 2020, reserving 10% of randomly shuffled datapoints as a test set. We first tested if the β-GMVAE model could adequately form discrete clusters of our market data. As a benchmark, we attempted to cluster our market data using PCA and a Gaussian Mixture Model(GMM). We choose GMM as our shallow clustering layer for our benchmark because, after the cubic transformation, each of our features nearly follows a standard normal distribution. We observed a significantly richer latent space, with clearly discrete clusters when using our β-GMVAE model as compared to PCA.

Figure 3 show the clusters found when applying our regime detection method on our test set of market data. Figure 4 shows the clusters formed using the first two principal components. The latent space created using the β-GMVAE model has discrete and separable clusters compared to a simple PCA transformation. Moreover, the latent space formed using the β-GMVAE is significantly less noisy than the PCA projection. Figure 5 shows the out of sample slippage results from using our regime detection method on our order data. It is clear that our regimes separate the orders in a fashion that also separates the orders' market impact. Figure 6 shows the out of sample slippage results using PCA and GMM. The benchmark collapses the out of sample order data into one cluster. This is non-optimal and demonstrates the robustness of a β-GMVAE for regime identification compared to traditional methods.

Using the regimes identified with our β-GMVAE, we analyze the out of sample during execution order features. Table 1 shows the average value for various order features during the execution of the order. We want to emphasize that these averages were computed across the order's horizon and not before the order. Moreover, the β-GMVAE only used features computed before the order to make the regime predictions. It is clear from these results that regimes persist during the execution of the order. It appears the

regimes stratify the different orders into various quote size, volatility, and illiquidity classes. Moreover, the slippage of each regime matches the previous relationships established in our previous paper. Orders where the quote size is large and volatility is low have low slippage, and orders where volatility is high and quote size is small have large slippage [7]. However, now we see two more regimes consisting of mid quote size orders. Moreover, the regimes of mid-quote size orders have very different market microstructure. One has very high volatility, and the other low volatility but high illiquidity. While both regimes have similar slippage, it is clear they have different variables driving the slippage. Hence, it is important to separate the regimes and execute differently in each regime.

TABLE 1
Order Feature Averages During Order Execution For Each Regime

Feature	Regime 1	Regime 2	Regime 3	Regime 4
Quote Size	14	40	87	32
Realized Volatility	5.4e-04	8.7e-04	5.1e-04	3.6e-04
Executed Quantity	7.9	20	18	12
Order Size	8.2	22	18	12
Amihud Illiquity	4.7	2.7	0.34	3
Volume During Execution	1.5e+03	1.3e+04	1.2e+04	3.2e+03
Arrival Price Slippage	0.50	0.27	0.059	0.23

CONCLUSION

In this paper we have introduced a novel application of deep generative models for execution regime identification. By leveraging a β-GMVAE we were able to easily identify various regimes of E-mini S&P futures. Unlike other methods, such as PCA+GMM our model does not suffer from mode collapse on an out of sample dataset. Additionally, we analyzed the market micro-structure of the various identified regimes and discovered each regime has a unique market micro-structure. Moreover, the different micro-structure results and their various aggregated market impact results were consistent with our previous findings. We believe our model is just scratching the surface at the possible applications of the β-GMVAE. We hope to leverage this model in future to predict regimes across instruments, leverage instrument similarities to predict regimes for newly introduced instruments, and serve as a basis for a cost mixture model.

References

[1] David Charte et al. “A Showcase of the Use of Autoencoders in Feature Learning Applications”. In: *Lecture Notes in Computer Science* (2019), pp. 412–421. ISSN: 1611-3349. DOI: 10.1007/978-3-030-19651-6_40. URL: http://dx.doi.org/10.1007/978-3-030-19651-6_40.

[2] Mark Collier and Hector Urdiales. *Scalable Deep Unsupervised Clustering with Concrete GMVAEs*. 2019. arXiv: 1909.08994 [cs.LG].

[3] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. The MIT Press, 2016. ISBN: 0262035618.

[4] I. Higgins et al. “beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework”. In: *ICLR*. 2017.

[5] Diederik P Kingma and Max Welling. *Auto-Encoding Variational Bayes*. 2014. arXiv: 1312.6114 [stat.ML].

[6] E. Min et al. "A Survey of Clustering With Deep Learning: From the Perspective of Network Architecture". In: *IEEE Access* 6 (2018), pp. 39501–39514. DOI: 10.1109/ACCESS.2018.2855437.

[7] Shankar Narayanan. *Volatility, Multidimensional Regimes and Execution Performance*. Quantitative Brokers. 23 March 2020.

[8] Alec Radford, Luke Metz, and Soumith Chintala. *Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks*. 2016. arXiv: 1511.06434 [cs.LG].

[9] Rui Shu. *GAUSSIAN MIXTURE VAE: LESSONS IN VARIATIONAL INFERENCE, GENERATIVE MODELS, AND DEEP NETS*. URL: http://ruishu.io/2016/12/25/gmvae/. (accessed: 12.30.2020).

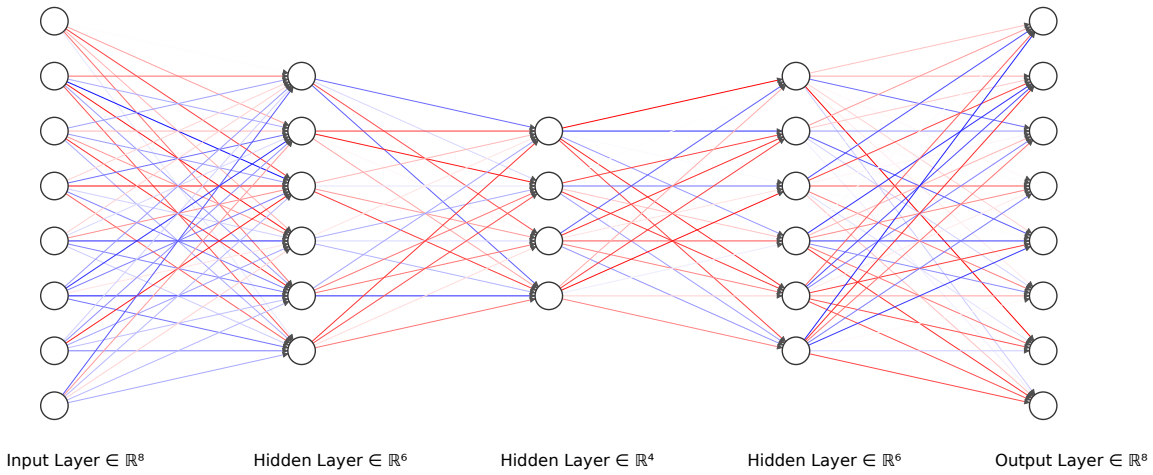


FIGURE 1. Figure shows a simple, fully connected autoencoder. The middle hidden layer of $\mathbb{R}^4$ is the latent "bottlenecked" space. We train this neural network end to end with the same input and output data. If autoencoder can sufficiently reconstruct the data, we can regard the middle latent space as a dense encoding of our original dataset. In many ways, this is a natural extension of Principle Component Analysis (PCA).

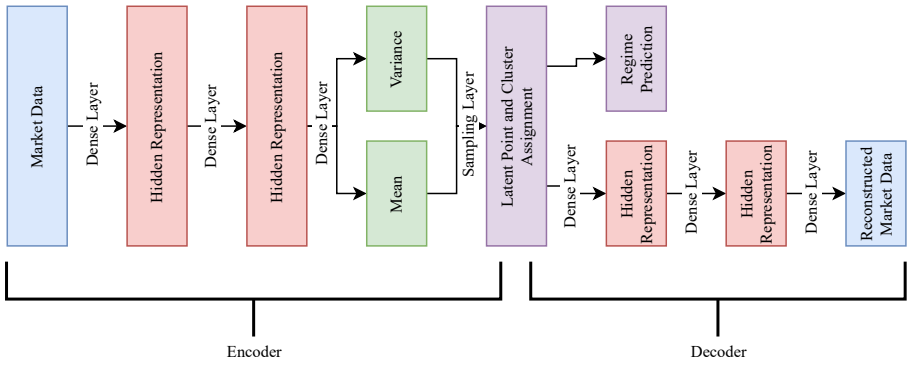
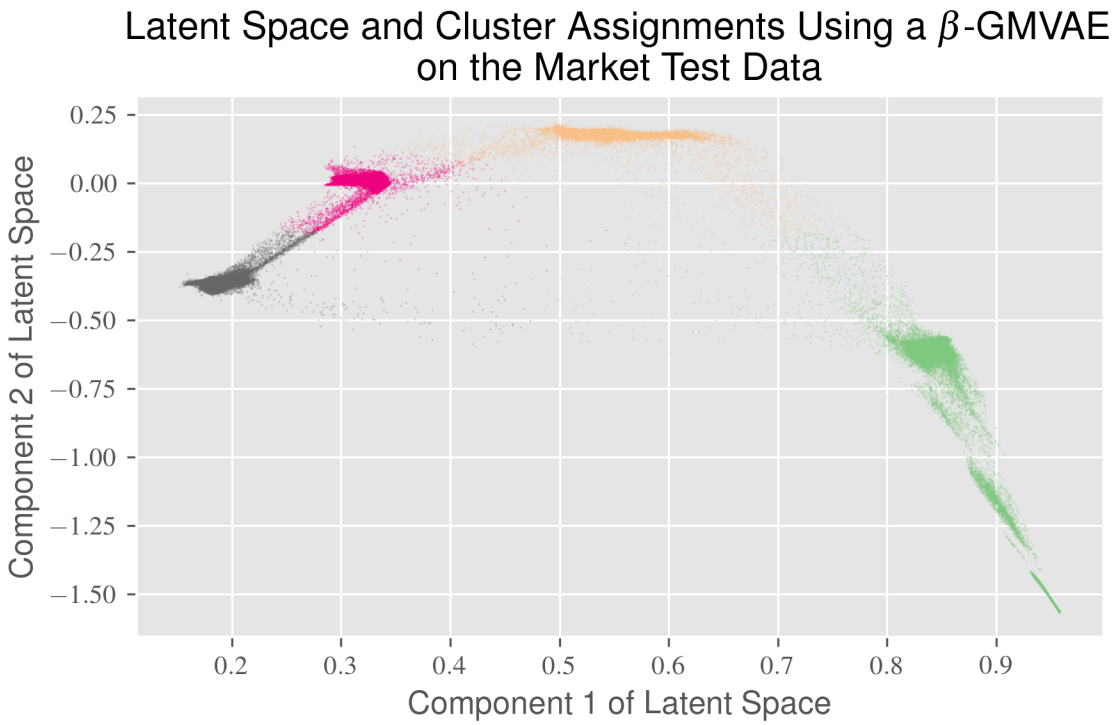


FIGURE 2. The regime identification model take in market data before an order, creates a latent representation using the trained encoder from the β-GMVAE and then uses the posterior category assignment as the cluster assignments.

FIGURE 3
Figure shows the embedding of the reserved market data test set using the β-GMVAE. In the figure we clearly see four well defined clusters of latent points. The coloring represent the various posterior cluster assignments assigned by the β-GMVAE. We see that the four populations are fairly balanced.



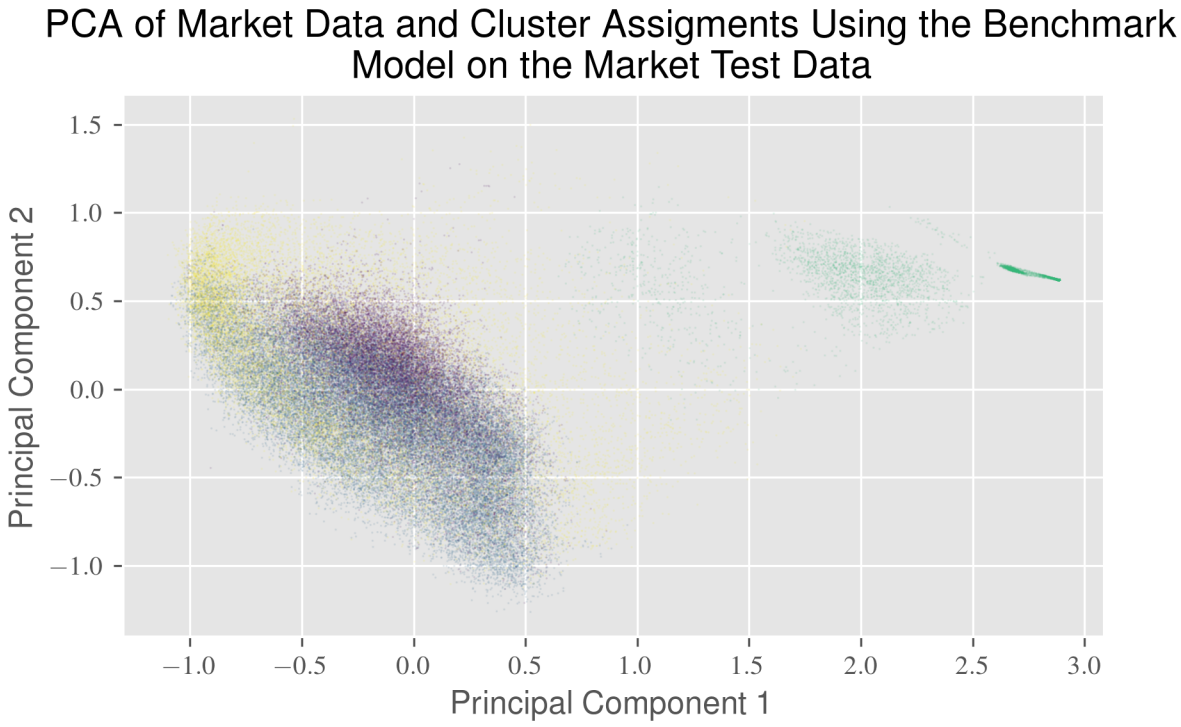
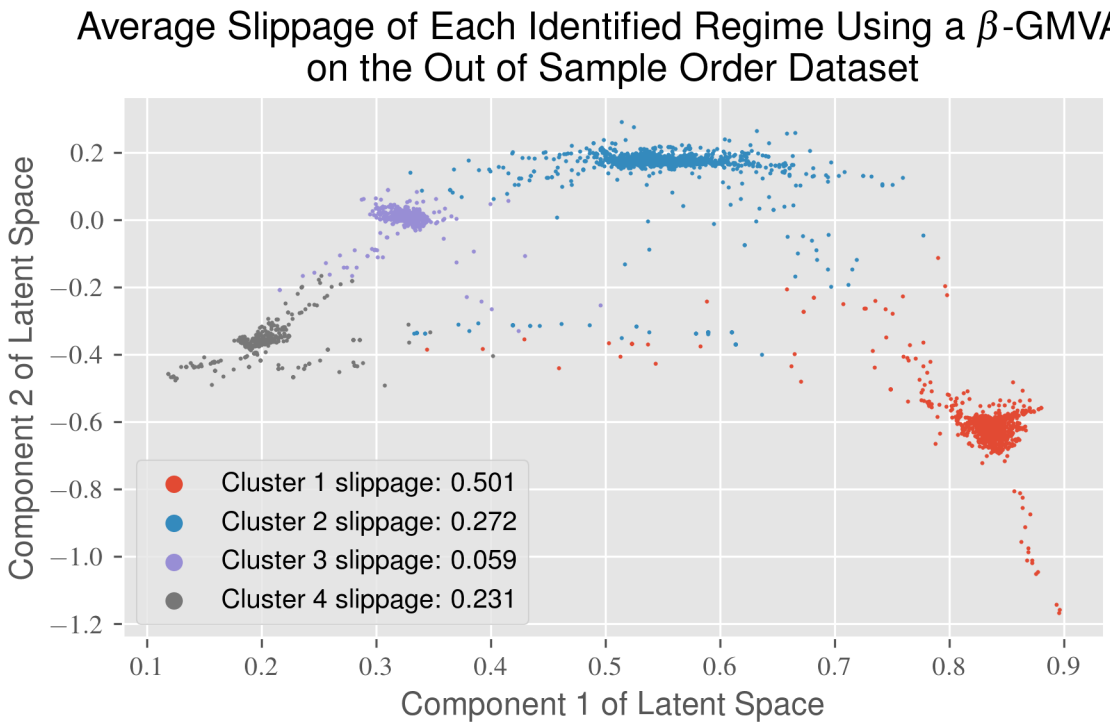


FIGURE 4. Figure shows the two dimensional PCA projection of the reserved market data test set and the cluster assignments provided by the GMM clustering layer. In the figure, we see only two clearly defined populations of data. In addition, we see that the GMM layer struggles to classify the four latent regimes of our data. Instead of identifying four equal sized populations, the mixture collapses the majority of the data into one cluster.

FIGURE 5
Figure shows the average slippage of our out of sample order data set within each of our identified regimes using the β-GMVAE. We see a clear separation of slippage characteristics among our various clusters. Moreover, even with our out of sample order data set the clusters remain balanced.



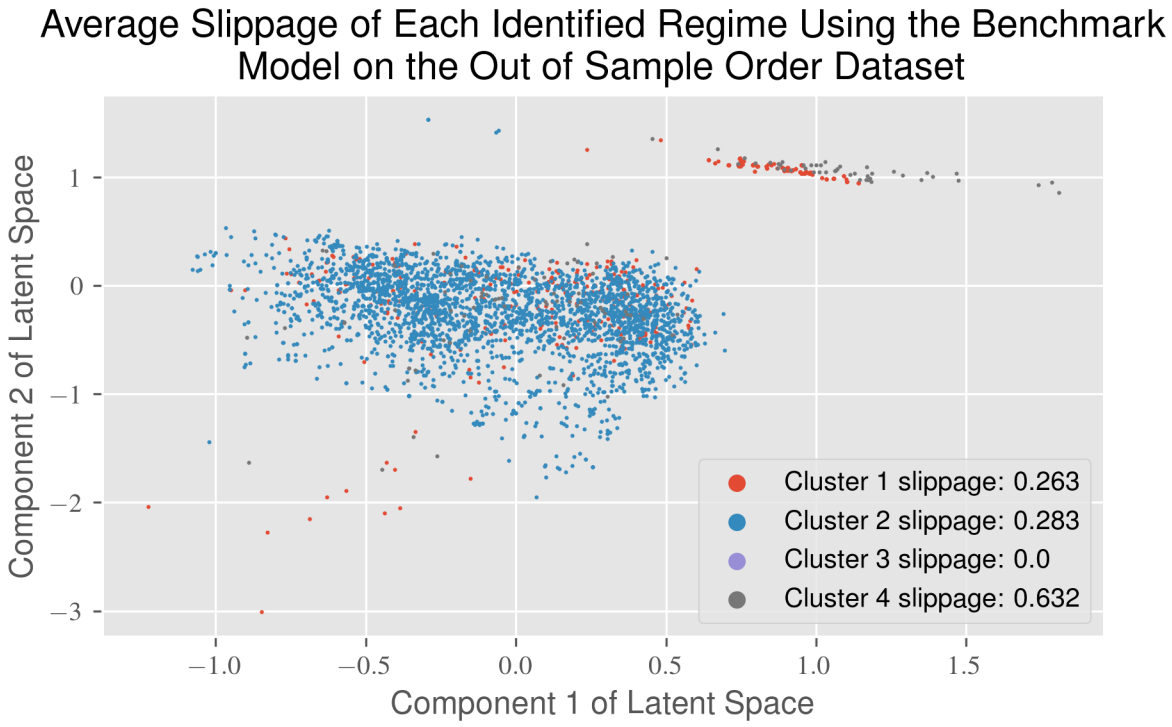


FIGURE 6. Figure shows the average slippage of our out of sample order data set with each of the identified regimes using the PCA+GMM benchmark model. Unfortunately, the entire latent space has collapsed into a single mixture component(with the exception of a few outliers) and hence it is clear this model provides an unusable and undesirable model for regime identification on our out of sample dataset.