HireVAE: An Online and Adaptive Factor Model Based on Hierarchical and Regime-Switch VAE

Zikai Wei\textsuperscript{1}, Anyi Rao\textsuperscript{2}, Bo Dai\textsuperscript{3} and Dahua Lin\textsuperscript{1,3}
\textsuperscript{1}The Chinese University of Hong Kong \textsuperscript{2}Stanford University \textsuperscript{3}Shanghai AI Laboratory
\{wz018, dhlin\}@ie.cuhk.edu.hk, anyirao@stanford.edu, daibo@pjlab.org.cn

Abstract

Factor model is a fundamental investment tool in quantitative investment, which can be empowered by deep learning to become more flexible and efficient in practical complicated investing situations. However, it is still an open question to build a factor model that can conduct stock prediction in an online and adaptive setting, where the model can adapt itself to match the current market regime identified based on only point-in-time market information. To tackle this problem, we propose the first deep learning based online and adaptive factor model, HireVAE, at the core of which is a hierarchical latent space that embeds the underlying relationship between the market situation and stock-wise latent factors, so that HireVAE can effectively estimate useful latent factors given only historical market information and subsequently predict accurate stock returns. Across four commonly used real stock market benchmarks, the proposed HireVAE demonstrate superior performance in terms of active returns over previous methods, verifying the potential of such online and adaptive factor model.

1 Introduction

Originating in asset pricing, factor model is the foundation of factor-based investing, which simplifies the high-dimensional characteristics in quantitative investment via an experimental way to model how factors drive returns and risks. This can significantly help institutional investors construct diverse and customized portfolios. But, a better-developed capital market brings out more challenges in achieving excess returns in the market using traditional methods due to more similar investing actions and overlapping trading positions. Although the recent nonlinear data-driven methods [Duan et al., 2022; Wei et al., 2022; Gu et al., 2021] enrich the conventional tool library built largely from linear methods [Ng et al., 1992; Fama and French, 2020], they still face serious issues in handling large, abrupt, and sustained market changes and some factors that worked well in the previous market situation become less effective in the new situation. Therefore, it is of great importance to model the current market regime and develop an adaptive factor model associated with different market situations.

Both practitioners and academics make many attempts to model the regime change in markets using statistical learning [Guidolin and Timmermann, 2007], machine learning [Uysal and Mulvey, 2021], or hybrid methods [Akioyamen et al., 2020] from different perspectives of the market. Existing data-driven approaches view regime switching as a clustering problem and represent each regime as a cluster, subdividing the market regime based on the information absorbed from the entire training data. However, this setting is not suitable for learning a regime-switching model intended for later use in investment practice. Practice requires an online learning setting that all data used for learning should be point-in-time data, meaning that there is no future information leakage at any timestamp in the training data set.

In this paper, we thus propose a novel end-to-end neural factor model, referred to as HireVAE, that offers an online and adaptive regime-switching capability, targeting two critical questions: \textbf{Q1:} How to keep the regime cluster labels changing consistently and centers of cluster shifting smoothly during training? \textbf{Q2:} How to adaptive learn regime-oriented factor model under different regimes?

To tackle Q1, we develop a module for identifying regimes, in which it learns a set of dynamic distributions associated with different regime clusters. To maintain this regime identifier as a stable clustering module, we design a linear stabilization algorithm to supervise the identifier learn a consistent clustering that smoothly alternates with a market latent variable. Specifically, a market encoder, which is learned in a prior-posterior learning framework, will encode the market information into a market latent variable. This framework can help the market encoder learn the hidden logic between the market information and the average future stock returns in the whole market, which will serve as prior knowledge for the market encoder in predicting future stock returns given only current market information. The market latent variable plays two roles in HireVAE: 1) at first it is used to estimate the means and variances of \( K \) Gaussian distributions, representing \( K \) regimes. 2) Subsequently, it will also be used to predict \( K \) regime indicators measuring the likelihood of each regime being the most suitable one, from which the one with the largest likelihood can thus be identified. It’s worth not-
To cope with assignment" is eliminated. Moreover, to smoothly update the regime and the additional burden of checking for a “wrong assignment” is eliminated. Also, we can always associate with a consistent regime clustering deep learning technique.

2 Related Work

Factor Models. The factor model are traditionally regarded as a pricing model in academia [Fama and French, 1992; Fama and French, 2020], while practitioners use its anomalies as factors to capture excess returns from mispricing [Levin, 1995]. In recent years, stock forecasting studies based on machine learning methods are emerging. Machine learning based stock prediction methods are closely related to factor models and can be categorized as factor mining and factor composition. The former learns a new logical factor from new data sources or perspective, while the latter finds a better way to compose the existing factors, which can be considered as a machine learning based factor model. In terms of factor mining, existing works on stock prediction study trading patterns [Wang et al., 2022; Ding et al., 2020; Zhang et al., 2017], investment reviews [Wang et al., 2021b], overnight patterns [Li et al., 2021], and the temporal relationship of stocks [Wang et al., 2021a]. As for factor composition as nonlinear factor models, existing works learn optimal latent factors via a variational autoencoder (Factor-VAE) [Duan et al., 2022], a deep multifactor model (DMFM), which builds on hierarchical stock graphs [Wei et al., 2022], and a deep risk factor model that fuses a set of uncorrelated risk factors from the original style factors [Lin et al., 2021]. Unlike existing methods, our model aims to learn an adaptive factor model that can recognize the current market regime and use a learnable and more appropriate factor composition for that market regime, which can enable a market-driven investment decision.

Regime Switching [Hamilton, 1989] provide a statistical modeling framework to address the problem of regime switching in various circumstances, such as regime switching in interest rates [Ang and Bekaert, 2002], asset allocation [Guidolin and Timmermann, 2007], and financial markets[Zhu, 2022]. In the cryptocurrency market, [Zhu, 2022] shows that regime switching methods perform better than methods without regime switching. Machine learning based methods open another main direction in modeling regime switching, such as Hidden Markov Model [Wang et al., 2020] and Gaussian Mixed Models [Botte and Bao, 2021]. As a hybrid method, [Akioyamen et al., 2020] applies principal component analysis and $k$-means clustering to identify regimes in financial markets. Deep learning based regime switching models of energy commodity prices [Mari and Mari, 2022]. Unlike the existing regime switching method, in investment practice, we do not need to use an online learning algorithm to identify market regimes, but also need to...
learn an adaptive factor model based on the determination of the market regime.

**Variational Autoencoder.** Variational Autoencoder (VAE) [Kingma and Welling, 2013], one of the major families of deep generative models, provides a prior-posterior framework that encodes the observation as prior knowledge in a latent space, and later [Vahdat and Kautz, 2020] introduce a hierarchical pipeline into VAE. [Chung et al., 2015] is the first to extend VAE to sequential modeling. Then, the variational methods are extended to neural machine translation [Su et al., 2018] and trajectory prediction. In the area of financial applications, VAE includes topics such as stochastic volatility models [Luo et al., 2018] and factor models. The latter VAE family includes applications in stock factor modeling based on social media in the context of stocks [Xu and Cohen, 2018], applications in factor composition where [Duan et al., 2022] aims to learn representative latent factors from existing technical factors, and [Gu et al., 2021] aims to learn a pricing model.

3 Preliminaries

In this section, we define the stock prediction problem and then introduce the problem setting from the perspective of learning a data-driven method.

**Definition 1 (Stock prediction).** In general, stock prediction problem can be defined as learning a mapping \( f (\mathcal{X}; \Theta) \) from the past information set \( \mathcal{F}_{\leq t} \) to future stock return \( y_{t+\Delta t} \), where \( \Delta t \) is the prediction length, \( y_{t+\Delta t} = \mathbb{E} (y_{t+\Delta t} | \mathcal{F}_{\leq t}) \) is its corresponding expected future return, and \( \Theta \) is the model parameter of data-driven method \( f (\cdot) \).

In our setting, we have a set of samples labeled with time stamps sorted in chronological order: \( D = \{D_0, \ldots, D_n\} \), where \( n \) is the total number of trading days covered by the dataset. For a single data sample \( D_t = \{X_t, M_t\} \), \( X_t \in \mathbb{R}^{T \times N \times C} \) and \( M_t \in \mathbb{R}^{T \times C} \) are sequential features of stocks and the market, where \( T, N \) and \( C_i \) represent the length of the historical sequence, the number of stocks and the number of features. Moreover, the overall pattern of the market (global observation) consists of information from \( d \) modalities, based on which it can be segmented as \( M_t = \{M_t^1, \ldots, M_t^d\} \). The above are general input dataset.

Besides, the corresponding target set of \( D \) is \( \mathcal{Y} = \{Y_t, \ldots, Y_{t+\Delta t}\} \), where \( Y_t = \{y_t, \ldots, y_{t+\Delta t}\} \), \( \Delta t \) is the length of prediction horizon, and \( y_{t+\Delta t} \in \mathbb{R}^N \) is the future cross-section returns of \( N \) stocks in the market over the time period from \( t \) to \( t + \Delta t \).

Our target is to learn an optimal \( \Theta^* \) of a data driven model \( f (\cdot; \Theta) \) with learnable parameter \( \Theta \) such that \( \hat{Y} \leftarrow f (D; \Theta) \).

4 Methodology

In this section, we introduce HireVAE to identify the market regime with online learning in conjunction with a linear stabilization clustering algorithm and learn an adaptive factor model to extract the latent logic behind the market and stocks under different regimes. The brief framework of HireVAE is given in Figure 2.

The architecture of our model is derived from a VAE encoder-decoder structure. In contrast to the classical pipeline, we develop a hierarchical encoder with a two-level latent space representing a market-stock hierarchy, and couple it with a set of regime-specific decoders that reconstruct or predict stock returns based on different market situations. First, the market encoder extracts market latent factors from multimodal features. Then, the stock encoder plays the role of a conditional encoder that extracts stock latent factors based on stock features and market latent factors. This hierarchical structure can help the stock latent factor better recognize the market situation. In the middle of this encoder-decoder architecture, a regime-switching module acts as a mediator to match an appropriate regime-specific decoder to the latent market situation.

Specifically, this architecture consists of four parts: an explanation of the feature extraction procedure from sequential data, an overview of the hierarchical encoder-decoder structure, a regime switching learning algorithm, and the learning objectives in prior-posterior learning.

4.1 Extracting Features from Sequential Data

We establish two inherent feature extractors, i.e., stock-specific and market-specific extractors.

**Stock Features.** The stock feature extractor \( \phi_{\text{feat}} \), extracts features \( e_s \) from the historical stock characteristics: \( e_s = \phi_{\text{feat}}(X) \), where \( X \in \mathbb{R}^{T \times N \times C} \) and \( e_s \in \mathbb{R}^{N \times H} \) represents the extracted stock features with a hidden size \( H \). Specifically, \( e_s \) is the latest hidden state obtained by a gated recurrent unit which summarizes all temporal information.

**Market Features.** For sequential market data \( M_t \in \mathbb{R}^{T \times C} \) consists of information with \( d \) multimodal features, we introduce a set of extractors \( \phi_{m,i} \) associated with different modalities \( M_t^i \in \mathbb{R}^{T \times C_i} \), \( i = 1, \ldots, d \), and \( \sum_{i=1}^d C_i = C \). We use \( d \) cross attention modules [Hou et al., 2019] to learn pairwise features from each modality pair. Later, we extract the market features \( v \) from all these pairwise features by concatenation. In our setting, a market can be represented by an index composed of all the stocks in the market. The different modalities are the market momentums (defined as the cumulative return over a period of time), the levels of relative trading volume, and the market volatilities over different time frames.

4.2 Hierarchical Encoder-Decoder Structure

To help the model better identify the market situation and better interpret future stock returns in different market situations, we design the latent space in a hierarchical structure that represents latent factors from the market and stocks. A pair of encoders is designed to learn latent variables from market and stocks based on the market and stock features.
Posterior Structure in Training Phase

In the training phase, the market encoder generates a market latent variable, based on which and in conjunction with both future stock returns and historical stock features a market-aware stock encoder can learn stock latent features. After the current market regime is identified by a regime switching algorithm, a regime-specific decoder can reconstruct future stock returns.

Market Encoder. The market encoder $\phi_m^{enc}$ extracts posterior market latent factor $m$ from the extracted market information $v$, and future average return of all stocks $y$:

$$[\mu_{post}^m, \sigma_{post}^m] = \phi_{enc}(v, y),$$

$$m \sim N \left( \mu_{post}^m, \text{diag}(\sigma_{post}^m) \right),$$

where $m \in \mathbb{R}^H$ is the market latent factor following a Gaussian distribution, parameterized by mean $\mu_{post}^m$ and standard deviation $\sigma_{post}^m$, and $H_m$ is the market hidden size.

Market-Aware Stock Encoder. The stock encoder $\phi_s^{enc}$ extracts posterior stock latent factors $z$ from the market latent factor $m$, stock features $e_s$, and future stock returns $y$:

$$[\mu_{post}^s, \sigma_{post}^s] = \phi_{enc}(v, e_s, y) = \phi_{s^{enc}}(v, e_s, y),$$

$$z \sim N \left( \mu_{post}^s, \text{diag}(\sigma_{post}^s) \right),$$

where $z \in \mathbb{R}^s$ is the stock latent factors following a Gaussian distribution, parameterized by mean $\mu_{post}^s$ and standard deviation $\sigma_{post}^s$, and $H_s$ is the stock hidden size.

Regime-Switch Factor Decoder. We develop a learning algorithm to distinguish the current market situation from different regimes in the latter part of the paper (Section 4.3). To learn an adaptive factor model for different market situations, we design a regime-switch factor decoder consisting of $N_k$ sub-decoders that can reconstruct future stock returns $\tilde{y}$ based on stock latent factors $z$ and market latent factors $m$:

$$\tilde{y} = \phi_{dec}(z, m, e_s; c),$$

where $c \in \{1, \ldots, N_k\}$ denotes a market regime. More precisely, $c = f(m)$ where $f(\cdot)$ is a switching function (algorithm) that can identify the regime based on the market latent factors $m$.

Prior Structure in Prediction Phase

First, the market prior generates market latent factors $m_0$ based on the extracted market information $v$ from various modalities. Later, the stock prior generates stock latent factors $z_0$ based on the market prior variables $m_0$ and stock characteristics $e_s$. The decoder predicts future returns based on the market-stock priors and the regime $c$ determined by the switch function. Specifically,

$$[\mu_0^m, \sigma_0^m] = \phi_{m^{prior}}(v),$$

$$m_0 \sim N \left( \mu_0^m, \text{diag}(\sigma_0^m) \right),$$

$$[\mu_0^s, \sigma_0^s] = \phi_{s^{prior}}(m_0, e_s),$$

$$z_0 \sim N \left( \mu_0^s, \text{diag}(\sigma_0^s) \right),$$

$$\tilde{y} = \phi_{dec}(z_0, m_0, e_s; c) \quad \text{with} \quad c = f(m_0),$$

where $m_0 \in \mathbb{R}^H$ follows a Gaussian distribution with parameters $\mu_0^m$ and $\sigma_0^m$, and $z_0 \in \mathbb{R}^s$ follows a Gaussian distribution with parameters $\mu_0^s$ and $\sigma_0^s$.

Algorithm 1 Linear stabilization clustering algorithm

**Input:** Training data $D^{train} = \{D_{\text{train}}^{\text{batch}}, \ldots, D_{\text{train}}^{\text{batch}}\}$ s.t. all data items are sorted in chronological order

**Parameter:** Model parameters $\Phi$, regime distribution parameters $\mu_r \in \mathbb{R}^N_r$ and $\sigma_r \in \mathbb{R}^N_r$.

**Output:** Regime $c_i, i = 1, \ldots, N_b$

1: Randomly initialize $\mu_r$ and $\sigma_r$ with $\sigma_r \geq 0$
2: while $i \leq N_b$
3: Projection to 1D score space: $s_i \leftarrow \text{Score}(D_{\text{train}}^{\text{batch}})$
4: Pred. cluster center: $\mu_i, \sigma_i \leftarrow \text{DistShift}(D_{\text{train}}^{\text{batch}})$
5: Sort cluster center: $\mu_i, \sigma_i \leftarrow \text{Sort}(\mu_i, \sigma_i)$
6: Update cluster center:
7: Calc. regime prob.: $p_i \leftarrow N \left( s_i; \mu_r, \text{diag}(\sigma_r^2) \right)$
8: Pred. regime: $c_i \leftarrow \text{ArgMax}(p_i)$
9: end while
10: return regime $c_i, i = 1, \ldots, N_b$

4.3 Regime Switching with Online Learning

In order to learn an adaptive factor model that can handle various regimes, the first step is to identify these regimes based on point-in-time (PIT) market information. To achieve consistent and smooth market clustering based on a market latent variable $m \in \mathbb{R}^H$, extracted from the mixed market features $v \in \mathbb{R}^H$, we develop an online learning clustering algorithm with linear stabilization summarized in Algorithm 1. First, we project the market latent variable onto a 1D measure space, and consider its relative score in this 1D space as a random variable from a market regime distribution. Then, we compute the log-likelihood of this score w.r.t. the distributions of different market regimes and choose the regime with largest log-likelihood as the predicted regime.

Projection to 1D Space. A project layer $\phi^{proj}$ projects the market latent variable $m \in \mathbb{R}^H$ onto a 1D regime measure space. We define $s = \phi^{proj}(m)$, $s \in \mathbb{R}$, as the “score” of the current market situation and it plays a role as a 1D measure of the overall market status. This market score $s$ is considered as a given sample in the sample spaces of the different regimes. Later, we can measure how much more likely this market score $s$ belongs to a given market regime.

Determination of Market Regimes. Denote the number of market regimes as $N_r$. We assume that $r$ is a Gaussian random vector given by $r \sim N \left( \mu_r, \text{diag} \left( \sigma_r \right) \right)$, where $\mu_r \in \mathbb{R}^N_r$ and standard deviation $\sigma_r \in \mathbb{R}^N_r$ are the representatives of regime centers and regime deviations w.r.t $N_r$ regime clusters. To keep each regime as different as possible, we use Kullback-Leibler divergence (KL) to measure the difference between two distributions and define the loss function in terms of market regimes as

$$L_{\text{reg}} = - \sum_{i=1, \ldots, N_r} \sum_{j=1, \ldots, N_r} \text{KL} \left( N \left( \mu_i, \sigma_i^2 \right) || N \left( \mu_j, \sigma_j^2 \right) \right),$$

where $\mu_i$ and $\sigma_i$ are the $i$-th element of the mean vector $\mu_r$ and the deviation vector $\sigma_r$, respectively.
Regime Learner. To determine the centers of each regime, the regime learner $\phi_{reg}$ predicts the parameters of the market regime distribution, i.e., $[\mu_r, \sigma_r] = \phi_{reg}(m)$. Specifically,

$$\mu_r = w_{\mu,r}v + b_{\mu,r},$$

$$\sigma_r = \text{Softplus}(w_{\sigma,r}v + b_{\sigma,r}).$$

To keep the shift of regimes consistent and smooth in 1D measure space, a sorting function is applied to the mean values of all regime centers $\mu_r \in \mathbb{R}^{N_r}$ in descending order and the corresponding deviation terms are reordered w.r.t. the new order of sorted mean values. Formally,

$$[\mu^*_r, I^*_r] = \text{Sort} (\mu_r),$$

$$\sigma^*_r = \sigma_r[I^*_r],$$

where $\mu^*_r$ is a reordered mean vector, $I^*_r$ is its corresponding reordered index w.r.t. its original index, and $\sigma^*_r$ is a reordered deviation vector w.r.t. this reordered index $I^*_r$. These reordered mean and deviation vectors obtained in each batch are later used to update each regime center ($\mu_r$ and $\sigma_r$) with a weighted moving average, allowing the regime center to be updated gradually.

Regime Prediction. We assume that the market score $s$ is a random variable generated from a market regime distribution defined as a Gaussian distribution. For $N_k$ regimes, we have $N_k$ market regime distributions, for each of which we can calculate the log-likelihood of the current market score $s$:

$$l(s; \mu^*_i, \sigma^*_i) = -\ln(\sigma^*_i) - \frac{1}{2} \ln(2\pi) - \frac{1}{2} \left( \frac{s - \mu^*_i}{\sigma^*_i} \right)^2,$$

where $i = 1, \ldots, N_k$. The regime is predicted to be the regime with the maximum log-likelihood among all regimes:

$$c = \arg\max_j [l(s; \mu^*_j, \sigma^*_j), i = 1, \ldots, N_k],$$

where $c$ indicates the predicted market regime.

4.4 Learning Objective

Our learning objective consists of three parts, where the first part is to learn a better future stock return reconstruction, the second part is to learn latent market features and latent stock features in a hierarchical structure, and the third part is to learn a more differentiated regime clusters. Thus, the overall loss function of our HiReVAE is as follows,

$$L_{rec} = -\sum_{r=1}^{N_r} \mathbf{1}_{f(m)(r)} \ln P_{\phi_{dec}}(\hat{y} = y|X, M, z, m; r)$$

$$L_{hier} = \text{KL}(P_{\phi_{enc}}(m|X, \hat{y})||P_{\phi_{prior}}(m|M, \hat{y})) + \text{KL}(P_{\phi_{enc}}(z|X, M, \hat{y})||P_{\phi_{prior}}(z|X, M, m))$$

$$L_{overall} = L_{rec} + L_{hier} + L_{reg}.$$

5 Experiments

In this section, we evaluate the proposed HiReVAE on real stock market data, choosing one of the largest emerging markets as representative, and demonstrate the effectiveness of our model through various experiments. These experiments are designed to investigate the following research questions. Exp 1: Does our method outperform the state-of-the-art methods in terms of factor investing? Exp 2: Is the improvement in effectiveness mainly due to the additional market information? Exp 3: How does regime switch help adaptive learning in factor models? Exp 4: Is the hierarchical structure necessary in our design?

5.1 Experiment Settings

Dataset Construction. We conduct the experiment over the entire China stock market, with price-volume and fundamental data obtained from the WIND database, and all raw data processed on a point-in-time basis. The stock pool is constructed based on all on-listing stocks, except for stocks that receive special treatments (ST) and those that have been on the market for less than three months. When a company is labeled as ST, which means it suffer losses for two or more consecutive years. We follow the same paradigm as in [Wei et al., 2022] to construct stock features and select 58 features that have a good coverage over the entire stock market. The length of the sequential data and the prediction horizon are $T = 20$ and $\Delta t = 20$, respectively. The stock returns are calculated on a basis of volume weighted average prices. We choose the dataset and data split settings as in [Wei et al., 2022] instead of [Duan et al., 2022] because the former better covers the period from 2010 to mid-2022 and has a longer test set from mid-2015 to mid-2022 than the latter from 2019 to 2020. There are 14 groups of training, validation, and test datasets. To prevent information leakage, we delete the data items from the validation dataset when the future stock returns in the validation dataset are covered by the time period of the test dataset.

Baselines. To allow a fair comparison, we provide the same sequential data to all methods. We compare our HiReVAE to the factor models as follows: ① Linear is a linear factor model derived from [Fama and French, 1992], with average pooling on the sequential dimension. ② GRU [Chung et al., 2014] is a factor model in which a GRU extracts the sequential features and, in conjunction with a linear layer, predicts future returns. ③ Trans [Ding et al., 2020] is a factor model.
based on Transformer. ④ MLP [Levin, 1995] is the first nonlinear factor model with multilayer perceptrons, where we equip it with a GRU to extract time-series features. ⑤ GAT [Veličković et al., 2018] is a graph attention work with a fully connected stock graph. ⑥ IGAT [Wei et al., 2022] is a GAT, based on a stock-industry graph where there is an edge between two nodes if two stocks belong to the same industry. ⑦ DMFM [Wei et al., 2022] is a deep multifactor model built on top of a stock graph. ⑧ VAE [Duan et al., 2022] is a stock prediction method based on VAE that treats latent stock variables as the optimal factors thus learned. ⑨ CVAE [Gu et al., 2021] is a factor model with a conditional autoencoder.

5.2 Exp 1: Effectiveness of HireV AE
In this experiment, we verify the effectiveness of our HireV AE with respect to factor investing. In this context, factor models are the essential tools in factor investing, where the objective is to answer two practical questions: 1) whether the model explains (future) stock returns well and 2) whether it provides good stock selection.

Predictive Power
To evaluate the predictive power of the compared methods, we use two metrics widely used in industry and academia [Duan et al., 2022; Wei et al., 2022], i.e., the rank information coefficient (IC), the rank information coefficient (RankIC), and the information ratio of RankIC (RankICIR), where RankICIR is the z-score of RankIC. To reconcile deterministic and generative methods in prediction, we use the means of the hierarchical priors as the deterministic values and determine the predicted stock return by \( \hat{y} = \phi_{dec}(\mu_0, \mu_m, e; c) \). By comparison in Table 1, HireV AE outperforms the compared methods, illustrating its effectiveness in stock prediction.

Stock Selection
The most widely used method to test the ability of a factor model in portfolio construction is Top-\(i\)-of-\(G\) Groups. Specifically, this involves sorting stocks in descending order \( w.r.t \) the expected returns estimated by a factor model and then dividing the sorted stocks into \( G \) groups. In addition, a long-short portfolio can be created by buying the Top-\(i\) group and short selling the Bottom-\(i\) group simultaneously. Since short selling in Chinese stock markets is not always possible for each stock, we construct only the portfolio holding the Top-\(1\) group. The frequency of rebalancing is monthly. All transaction costs and taxes are included in our backtesting.

The CSI300, the CSI500, the CSI1000 and the CNI2000 are widely used benchmarks in China stock markets and consist of the most representative stocks in market hierarchy. In particular, the CSI300 consists of the 300 most liquid and largest stocks. The CNI500, representing mid-caps, consists of the 500 most liquid and largest stocks. The CNI1000 consists of the following 1000 stocks representing small caps. In addition, the CNI2000 is another representative of small caps and consists of 2000 stocks.

To test the potential of subsequent use in the construction of enhanced index funds (EIFs), we compare the different methods. We report two metrics that are widely used in investment practice: the annualized active return (AR) relative to an investment benchmark and the information ratio (IR), the latter being a risk-adjusted measure. The “-s” stands for the exclusive use of stock characteristics.

Table 2: The performances on portfolio construction over the period from mid-2015 to mid-2022 (best/2nd best).

<table>
<thead>
<tr>
<th>Method</th>
<th>CSI 300 AR↑</th>
<th>CSI 300 IR↑</th>
<th>CSI 500 AR↑</th>
<th>CSI 500 IR↑</th>
<th>CSI 1000 AR↑</th>
<th>CSI 1000 IR↑</th>
<th>CSI 2000 AR↑</th>
<th>CSI 2000 IR↑</th>
<th>CSI ALL AR↑</th>
<th>CSI ALL IR↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear</td>
<td>11.94</td>
<td>0.30</td>
<td>8.81</td>
<td>0.50</td>
<td>6.98</td>
<td>0.65</td>
<td>5.04</td>
<td>0.49</td>
<td>5.82</td>
<td>0.89</td>
</tr>
<tr>
<td>GRU</td>
<td>19.40</td>
<td>0.60</td>
<td>15.04</td>
<td>0.80</td>
<td>12.89</td>
<td>0.96</td>
<td>11.27</td>
<td>0.84</td>
<td>13.25</td>
<td>2.20</td>
</tr>
<tr>
<td>MLP</td>
<td>23.04</td>
<td>0.78</td>
<td>20.20</td>
<td>1.10</td>
<td>18.12</td>
<td>1.22</td>
<td>16.87</td>
<td>1.09</td>
<td>19.25</td>
<td>3.53</td>
</tr>
<tr>
<td>Trans</td>
<td>7.21</td>
<td>0.18</td>
<td>4.63</td>
<td>0.28</td>
<td>3.03</td>
<td>0.41</td>
<td>0.68</td>
<td>0.24</td>
<td>0.22</td>
<td>0.27</td>
</tr>
<tr>
<td>GAT</td>
<td>5.77</td>
<td>0.14</td>
<td>5.65</td>
<td>0.19</td>
<td>4.33</td>
<td>0.29</td>
<td>2.30</td>
<td>0.16</td>
<td>2.08</td>
<td>0.31</td>
</tr>
<tr>
<td>IGAT</td>
<td>12.67</td>
<td>0.41</td>
<td>10.32</td>
<td>0.61</td>
<td>8.47</td>
<td>0.69</td>
<td>6.85</td>
<td>0.54</td>
<td>8.43</td>
<td>2.04</td>
</tr>
<tr>
<td>DMFM</td>
<td>3.16</td>
<td>0.05</td>
<td>2.13</td>
<td>0.20</td>
<td>0.78</td>
<td>0.35</td>
<td>1.50</td>
<td>0.18</td>
<td>2.24</td>
<td>0.32</td>
</tr>
<tr>
<td>VAE</td>
<td>26.49</td>
<td>0.82</td>
<td>25.04</td>
<td>1.11</td>
<td>23.32</td>
<td>1.23</td>
<td>19.05</td>
<td>1.02</td>
<td>23.72</td>
<td>4.38</td>
</tr>
<tr>
<td>CV AE</td>
<td>27.78</td>
<td>0.91</td>
<td>25.84</td>
<td>1.24</td>
<td>24.00</td>
<td>1.36</td>
<td>22.76</td>
<td>0.99</td>
<td>24.91</td>
<td>5.46</td>
</tr>
<tr>
<td>HiReV AE</td>
<td>30.89</td>
<td>0.89</td>
<td>29.05</td>
<td>1.23</td>
<td>27.14</td>
<td>1.35</td>
<td>25.94</td>
<td>1.18</td>
<td>28.02</td>
<td>5.92</td>
</tr>
</tbody>
</table>

Table 3: Ablation study of the difference in the use of market information. RIC rICIR, AR and IR are abbreviations for rank IC, rankICIR and active return, portfolio information ratio, respectively. The “-s” stands for the exclusive use of stock characteristics.

<table>
<thead>
<tr>
<th>Method</th>
<th>Mkt</th>
<th>rIC</th>
<th>rICIR</th>
<th>AR</th>
<th>IR</th>
</tr>
</thead>
<tbody>
<tr>
<td>GRU</td>
<td>✓</td>
<td>0.051</td>
<td>0.484</td>
<td>13.180</td>
<td>2.202</td>
</tr>
<tr>
<td>GRU-s</td>
<td>×</td>
<td>0.050</td>
<td>0.485</td>
<td>13.249</td>
<td>2.107</td>
</tr>
<tr>
<td>MLP</td>
<td>✓</td>
<td>0.051</td>
<td>0.537</td>
<td>19.247</td>
<td>3.525</td>
</tr>
<tr>
<td>MLP-s</td>
<td>×</td>
<td>0.051</td>
<td>0.540</td>
<td>19.251</td>
<td>3.454</td>
</tr>
<tr>
<td>VAE</td>
<td>✓</td>
<td>0.059</td>
<td>0.539</td>
<td>23.724</td>
<td>4.377</td>
</tr>
<tr>
<td>VAE-s</td>
<td>×</td>
<td>0.059</td>
<td>0.540</td>
<td>23.589</td>
<td>4.267</td>
</tr>
</tbody>
</table>

5.3 Exp 2: Improvement Not From Extra Info
Theoretically, see our experiment settings in Section 5.1, we use all stocks included in the CSI All A Share Index, an overall market representative of Chinese stock markets. At its core, the index is roughly calculated by a capitalization-weighted sum of the prices of all stocks, which means that...
this index is calculated based on the most basic stock characteristics, price and market capitalization. Some of the original stock characteristics are also calculated based on these two basic characteristics. Thus, the market index is a “global observation” of all stocks and no additional information is added to the original raw data.

To provide empirical support for the previous argument, we conduct two experiments. First, all methods use the same input characteristics for stocks and the market, and the corresponding results are reported in Table 1 and 2. Second, in Table 3, we show the comparisons between the baseline methods in terms of using market information, selecting the baseline methods that performed well in Exp 1 as our baseline methods in Exp 2. The results show that the additional market information does not help to improve the performance of a model.

5.4 Exp 3: Adaptive Learning is Better

Determining the regime on the basis of volatility is a way of establishing regimes on the basis of rules of thumb [Botte and Bao, 2021]. Inspired by this knowledge and experience, we develop a rule-based method shown in Algorithm 2. We consider this method as a baseline for regime switching to its online and adaptive counterpart. Table 4 shows the performance of the different regime-switching algorithms in stock prediction and portfolio construction, where we select the most recommended stocks from the entire market (10% stocks from the market). In different modeling settings, our adaptive regime switching algorithm can always outperform the rule-based and neural clustering counterparts.

5.5 Exp 4: Hierarchy is Necessary

In this experiment, we analyze the necessity of using a hierarchical structure in two ways: first, does a hierarchical latent space (HiVAE) encode market information better than a single latent space (VAE)? Second, does the hierarchical structure help the end-to-end clustering algorithm in adaptive learning of market regimes.

Table 5 can answer these two questions accordingly. As for the first question, encoding market information alone does not provide significant improvement in the non-regime-switching modeling, since the metrics of VAE and HiVAE are similar. This is reasonable, as we showed in Section 5.3, where the market information itself does not provide more information than the raw data. However, in the case of regime switching, the market latent variable plays an important role in end-to-end regime switching learning, as it is used to estimate the regime region and predict the corresponding indicators that measure the probability of being in each regime. This can be empirically demonstrated by the fact that HireVAE outperforms the other two regime switching methods without considering the market latent variable, RVAE and RCVAE, in terms of predictive power and stock selection ability.

6 Conclusion

In this paper, we present HireVAE, a novel end-to-end neural factor model that can identify current market regime according to point-in-time market information, and subsequently adapt itself for better prediction. HireVAE achieves such an ability by a pair of hierarchically organized encoders respectively process global market situation and stock-wise latent factors. Being the first online and adaptive regime-switch factor model, HireVAE achieves superior performance on real stock market data, outperforming classical linear methods and recent non-linear data-driven methods by a large margin.
Acknowledgments

This project is funded in part by Shanghai AI Laboratory, CUHK Interdisciplinary AI Research Institute, and the Centre for Perceptual and Interactive Intelligence (CPII) Ltd under the Innovation and Technology Commission (ITC)’s InnovHK.

References


