Coupling Macro-Sector-Micro Financial Indicators for Learning Stock Representations with Less Uncertainty

Guifeng Wang, [1] **Longbing Cao,** [2] **Hongke Zhao,** [3] **Qi Liu,** [1] **Enhong Chen** [1*]

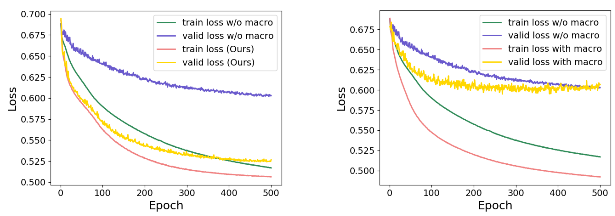
[1] Anhui Province Key Laboratory of Big Data Analysis and Application, School of Computer Science, University of Science and Technology of China, [2] Advanced Analytics Institute, University of Technology Sydney, [3] Tianjin University
wgf1109@mail.ustc.edu.cn, longbing.cao@uts.edu.au, hongke@tju.edu.cn, {qiliuql, cheneh}@ustc.edu.cn

Abstract

While the stock movement prediction has been intensively studied, existing work suffers from weak generalization because of the uncertainty in both data and modeling. On one hand, training a stock representation on stochastic stock data in an end-to-end manner may lead to excessive modeling, which involves the *model uncertainty*. On the other, the analysis of correlating stock data with its relevant factors involves the *data uncertainty*. To simultaneously address such uncertainty both from data and modeling perspectives, a fundamental yet challenging task is to learn a better stock representation with less uncertainty by considering hierarchical couplings from the macro-level to the sector-and micro-level. Accordingly, we propose a copula-based contrastive predictive coding (Co-CPC) method. Co-CPC first models the dependence between a certain stock sector and relevant macroeconomic variables that are sequential and heterogeneous, e.g., macro-variables are associated with different time intervals, scales and distributions. Then, by involving a macro-sector context, stock representations are learned in a self-supervised way that can further be used for downstream tasks like stock movement prediction. Extensive experiments on two typical stock datasets verify the effectiveness of our Co-CPC method.

Introduction

Predicting stock price movement continuously attracts interest in particular in advanced machine learning including deep learning (Lin, Guo, and Aberer 2017; Cao 2020) However, the highly volatile and non-stationary nature of stock markets often challenges the generalization of stock price predictors. As shown in Figure 1(a), the training and validation loss scores of two models show different trends. Obviously, a model with good generalization means its validation loss follows the decreasing trend of training loss like of the Co-CPC model rather than the LSTM one with very little decrease. The main reason behind it is that the encoder-decoder-based model directly adopts the stochastic variables as input and the back-propagation loss also consists of future uncertain data. That means the encoder and decoder simultaneously depend on the uncertain data (we call it *model uncertainty*), which may drag down the improvement from training examples to unknown validation examples.



(a) Co-CPC and LSTM (b) LSTM with varied inputs

Figure 1: The training and validation performance of two models on stock movement prediction. Noted that the predictor structure of Co-CPC and LSTM are the same.

In addition, recent efforts are made on incorporating relevant explicit and implicit financial indicators, e.g., trading behaviors (Cao, Ou, and Yu 2012; Vo and Phan 2019), social media data (Sun et al. 2017; Ding et al. 2016), and other financial indicators or markets (Gao et al. 2016; Cao, Hu, and Cao 2015), to model their influence on stock price movement. To some extent, these additional indicators can be seen as an attempt to mitigate the randomness of stock prices at the micro-level from the data perspective. While the mitigation of uncertainty both from data (i.e., further consider the macro-level) and model perspectives has not been studied.

From the data perspective, although the micro-level indicators are more relevant to the fluctuation of stock prices thanks to their fast updating speed, it tends to capture the short-term and local influence relationships, which is an advantage but can also suffer from overlooking the partial relevance. To deal with that, it is necessary to consider the effect both from micro-level and macro-level aspects. In economics, fundamental macroeconomic variables such as exchange rate, interest rate, industrial production and inflation are believed to be associated with or influence stock prices (Adam and Tweneboah 2008; Beber and Brandt 2008). Although economic studies have empirically shown the necessity to consider macro-data like macroeconomic indicators in stock price movement (Chen 2009), much less work has been on automatically modeling the complicated couplings between the indicators and stocks (Cao 2015; Cao, Hu, and Cao 2015). Intuitively, we can directly stitch macro-

*Corresponding author: Enhong Chen

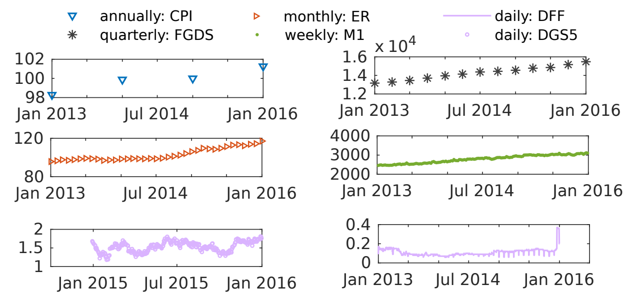


Figure 2: Some macro-factors to show their varied characteristics (i.e., time intervals, scales and distributions). The abbreviation of macro-factors could refer to the section on Experiments for more details.

variables as features with micro-ones to predictors, the results are also shown in Figure 1(b). Even if macro-level variables are further considered, the prediction performance of the direct splicing method is not significant because of its relatively limited ability of modeling the macro-micro variable couplings (Cao 2020) (named *data uncertainty*).

However, alleviating the uncertainty issues in both data and modeling of stock movement is a non-trivial task owing to multifaceted data characteristics and corresponding modeling challenges. First, as shown in Figure 2, multiple macroeconomic indicators are heterogeneous and unaligned, each may be acquired at a specific time interval (e.g., daily, weekly or monthly), have distinct stylized fact, hold varied distributions, and evolve over time at the macro-level. Second, macroeconomic variables (macro-level) and stock price series (micro-level) are generally sequential but unaligned with each other in terms of their values, granularities, intervals and distributions, etc. Last but not least, the better prediction performance requires to separate the learning process of encoder and decoder modules because the encoder directly learned based on the prediction loss fails to learn a better stock representation for the decoder (predictor).

Motivated by the above challenges and gaps, this paper aims to enhance the performance of modeling stock price movement by mitigating both data and modeling uncertainty. We consider the following settings to address the aforementioned data characteristics and model challenges: 1) modeling the couplings between heterogeneous macroeconomic factors by considering their respective data characteristics; 2) modeling the connection and influence of macro-factors with stock movement; and 3) representing stock states in a self-supervised way to avoid interference with random data when back-propagating to the encoder.

Accordingly, this paper introduces an unsupervised *Co*pula-enhanced *C*ontrastive *P*redictive *C*oding (Co-CPC) model to represent the couplings and influence between heterogeneous macroeconomic variables and stocks and to further estimate price movement. Specifically, simply applying copula to model the dependence cannot handle the misalignment and heterogeneity between variables and their sequential dynamics. Instead, we introduce a deep model into copula to model the couplings and influence between macro and micro variables, which preserves the respective characteristics of these variables and depresses the noise caused by their misalignment. On the basis of learned relations and influence, the macro-context of stocks is learned by our designed gating module. To uncover the impact of macro-variables and build a generalized stock representation with less uncertainty, we then apply the CPC (Oord, Li, and Vinyals 2018) method to mitigate the model uncertainty of the self-supervised learning.

Our work represents a new attempt of integrating copula with deep learning to couple macro and micro variables while handling their heterogeneities, interactions and influence. It also provides a new view on dealing with the data uncertainty of stock price movement task. Besides, in order to avoid the interference of random data to model learning, we try to learn a better encoder function through designing a distinct task from the target task. The experiments on two widely-used datasets evaluate the performance of our Co-CPC model.

Related Work

There are some classic methods focus on modeling correlation among multiple variables or time series. These methods include hypothesis test, multivariate analysis (Ibrahim 1999; Leontjeva and Kuzovkin 2016; Stefani et al. 2019), dependence learning such as copula-based high-dimensional dependence modeling (Choroś, Ibragimov, and Permiakova 2010), and transitional models such as coupled hidden Markov models (Cao, Ou, and Yu 2012; Chatigny et al. 2018). However, these methods are incapable of modeling the aforementioned heterogeneous and complex characteristics, let alone various macro-micro interactions (Cao 2020).

Generally, more and more deep models are used for time series prediction task, such as recurrent neural networks like LSTM (Nti, Adekoya, and Weyori 2019) and convolution neural network (CNN) based method like TCN (Bai, Kolter, and Koltun 2018), which are evidenced beneficial for capturing effective historical information of single time series. As we know that multiple time series are inherently temporally related in reality, independently modeling the property of each time series would lose some information about cross temporal coupling. Actually, Qin et al. (2017) propose a dual-stage attention based RNN method to select relevant driving series to make prediction by considering the effect of other time series via attention mechanism but it requires that all time series be aligned and homogeneous.

Recent researchers already made good attempts on stock movement forecasting. Some fundamental analysis related work focuses on exploring additional factors except stock price features. For instance, Xu et al. (2018) incorporate signals from social media, using public opinions to assist in digging the future fluctuation of stocks. What's more, Zhang et al. (2018) further consider the impact of news events on investment decisions. Specifically, they adopt coupled matrix and tensor factorization to expose the correlation among stocks. In a nutshell, such kind of works can be seen as easing the uncertainty issue in stock forecasting from a data point of view.

From the model aspect, most of the studies assume that the stock price is stationary but neglect the uncertainty that its random nature brings to the model. StockNet (Xu and Cohen 2018) and Adv-ALSTM (Feng et al. 2019) are exception which tackle such uncertainty from model perspective. StockNet applies VAE to encode the inputs into a latent distribution then enforce samples from it to be decoded with the same prediction. While Adv-ALSTM leverages adversarial training with attention based LSTM method to improve the model generalization. These two models are similar in mitigating the uncertainty by adding stochastic perturbation to the encoders. Different from them, our model tackles the uncertainty both from data and modeling perspectives respectively reflected on the incorporated macro-micro couplings and self-supervised representation model.

Preliminary

Problem Definition

Given m macroeconomic indicator series, i.e., $\mathbf{Q} = (\mathbf{q}_1, \mathbf{q}_2, ..., \mathbf{q}_m)$, where each indicator has its own time interval τ (as shown in Figure 2), hence for heterogeneous multiple time series, we represent the i-th indicator of length T_i as $\mathbf{q}_i = [q_{i,1}, ..., q_{i,t\tau}, ..., q_{i,T_i}]$.

Typically, given the previous target micro (stock) time series $\mathbf{y} = [y_1, y_2, ..., y_T]$ and its D-dimension features (e.g., open, high, low, close prices), i.e., $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_T] \in \mathbb{R}^{D\times T}$, as well as m macro-indicator series $\mathbf{Q}$, our aim is to learn an encoder function g_{enc} to represent stock well, further for subsequent forecasting services:

$$\begin{aligned} g_{enc} &= F(\mathbf{Q}_{\leq T}, \mathbf{y}_{\leq T}, \mathbf{X}_{\leq T}) \\ \hat{y}_{T+1}, ..., \hat{y}_{T+k} &= g_{dec}(g_{enc}(\mathbf{x}_{T+1}, ..., \mathbf{x}_{T+k})) \end{aligned} \tag{1}$$

where $F(\cdot)$ is a model to learn encoder function, then a predictor (decoder function g_{dec}) is applied to forecast. Our model follows encoder-decoder framework but each part are trained separately. Especially, the encoder function are learned without based on specific stock future labels, for mitigating the disturbation of future stochastic data.

Copula

An n-dimensional *Copula* function $C : [0, 1]^n \to [0, 1]$, is a joint cumulative distribution function (CDF) of a set of marginally random variables $\mathbf{u} = (u_1, ..., u_n)$. Sklar's theorem (Sklar 1973) provides the theoretical foundation for copulas, which states that every multivariate CDF could be decomposed into its marginals F_i and a unique copula C:

$$F(\mathbf{u}) = C(F_1(u_1), ..., F_n(u_n)) \tag{2}$$

There are two main reasons why we apply copula in our framework. First, the decomposition way allows us to separately estimate the univariate marginal distributions and the dependence structure. Hence, we can extend copula methods to learn the marginal distributions when information is missing and misaligned. Second, the copula is invariant to the univariate marginal distributions, that means the dependence structure is unchanged if we apply strict increasing transformations to the variables. This trait allows us to learn stable associative relationships even when variables are transformed into latent space.

In a nutshell, a copula characterizes the association within the latent random vector in the normalized space, decoupled from the possibly complex marginal-specific distributions.

Methodology

The core of Co-CPC model is to alleviate the uncertainty of micro changes from two perspectives. On the one hand, from the data uncertainty view, we use the relationship between macro-variables and stock sectors to guide the general trend of stocks in the follow-up. On the other hand, by designing a self-supervised task with fusing the macro-sector context, it is beneficial to explore the unique inner nature of micro time series from the model uncertainty view. Therefore, the whole macro-sector-micro framework is naturally divided into two parts, i.e., learning macro-context coupled with a specific stock sector (Figure 3), and representing stocks under macro-sector context in a self-supervised way (Figure 4).

Macro-sector Context Learning

Pipeline To explore how macro-factors affect future stock movement, we first use the stock sector as a medium to model the macro-micro coupling relation and then generate the macro-sector context representation for a stock. As shown in Figure 3, to deal with the heterogeneity and misalignment among variables, each observation sequence is mapped to a hidden space to learn its marginal distribution. Then a copula function gathers them together to learn a joint distribution with coefficient matrix $\mathbf{R}$. By considering such a relation, a macro-sector context for specific stock could be generated by a gating function.

In particular, a stock's price usually exhibits similar fluctuation trends as its stock sector. Compared with a single stock, the impact of macro-factors on a certain stock sector is more stable and easy to capture. As illustrated before, copula is quite suitable for modeling the nexus between heterogeneous macroeconomic variables and one specific stock sector. A common modeling choice for C is to use the Gaussian copula, given m sequential macro-variables $\mathbf{q}_1, ..., \mathbf{q}_m$ and a sequential stock sector variables $\mathbf{q}_0$, it is defined as:

$$C(\cdot) = \Phi_{\mathbf{R}}(\Phi^{-1}(F_0(q_0)), ..., \Phi^{-1}(F_m(q_m))) \tag{3}$$

Here, stock sector series is the average daily adjusted close price of the stocks belonging to this sector. The symbol $\Phi^{-1}(\cdot)$ is the inverse cumulative distribution function of a standard normal and $\Phi_{\mathbf{R}}(\cdot)$ is the joint cumulative distribution function of a multivariate normal distribution parametrized with mean vector zero and covariance matrix $\mathbf{R} \in \mathbb{R}^{(m+1)\times(m+1)}$. In brief, the dependence structure of these indicators is reflected in a Gaussian distribution.

Existing estimation methods for learning copula function (Choroś, Ibragimov, and Permiakova 2010), like parametric and semi-parametric estimation methods, require that the observation data is aligned on each sample. Since we involve unaligned variables where each variable $\mathbf{q}_i$ is observed in different time interval, resulting in different sequential lengths for variables. In addition, as multiple time series are not only time-varying but also heterogeneous, simply filling the missing value with the value of the last moment or

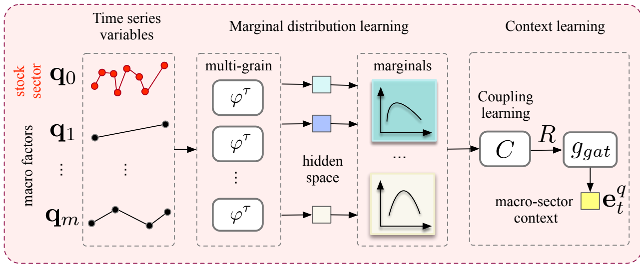


Figure 3: The framework of macro-sector context learning.

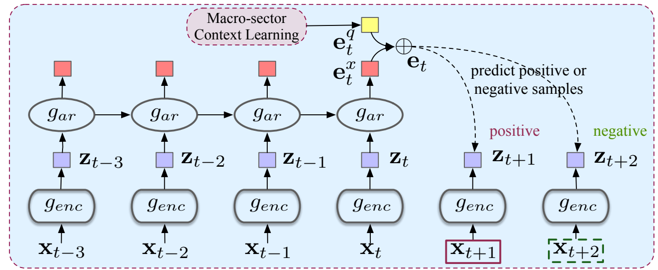


Figure 4: The framework of stock representation learning.

others would introduce noise, we thus cannot use the existing marginal estimation models. Instead, to preserve their own characteristics and relieve the noise caused by misalignment, we estimate marginal distribution $\hat{F}_i(\mathbf{q}_i; \boldsymbol{\beta}_i)$ for i-th variable by learning its parameter $\boldsymbol{\beta}_i$.

Marginal distribution learning in Multi-grain way. For time series variables, it is natural to learn the distribution parameters according to their cumulative states. While one may obtain the present representation by involving previous states with an RNN, we adopt a multi-grain way to learn their states with distinct time spans, for exploring the interaction of macro-indicators of varied granularities in the whole macro-context. Specifically, our model takes the form of a non-linear, deterministic state space model whose state $\mathbf{h}^{\tau}_{i,t} \in \mathbb{R}^{d_h \times 1}$ evolves independently for each time series $\mathbf{q}_i$ according to transition dynamics φ^{τ}:

$$\begin{aligned} \mathbf{h}^{\tau}_{i,t} &= \varphi^{\tau}(\mathbf{h}^{\tau}_{i,t-1}, q_{i,t\tau}, \mathbf{p}_t; \theta_h), \quad i = 0, 1, ..., m \\ \mathbf{p}_t &= Lookup(\kappa(t)) \end{aligned} \tag{4}$$

where the input of transition dynamics φ^{τ} contains the sequential variables and their specific time, which the value of day of week and day of year (denoted by $\kappa(t)$) are embedded in a lookup way. The transition dynamics are parametrized by a multi-grained LSTM with θ_h. The multi-grain way means that time series with the same time interval τ share the same parameters θ_h. For example, weekly variables ($\tau = 7$) share the same parameters and the last time step state $\mathbf{h}^{\tau}_{i,t-1}$ comes from 7 days ago.

Without loss of generality, we assume these observations satisfy Gaussian distribution (Salinas et al. 2019) whose mean and variance are derived from state $\mathbf{h}^{\tau}_{i,t}$, which is conductive to preserve their own specialty. It is formulated as:

$$\begin{aligned} \mu_{i,t} &= \mathbf{w}_{\mu}^{\top}\mathbf{h}^{\tau}_{i,t} \\ \sigma_{i,t} &= \mathbf{w}_{\sigma}^{\top}\mathbf{h}^{\tau}_{i,t} \end{aligned} \tag{5}$$

where $\mathbf{w}_{\mu}, \mathbf{w}_{\sigma} \in \mathbb{R}^{d_h \times 1}$ are parameters. Note that the distribution of each time series only depends on their own state $\mathbf{h}^{\tau}_{i,t}$, but the parameters are tied cross other time series, which benefits for subsequent joint distribution learning.

Learning the Macro-sector Context. Based on the learned marginal distribution $\hat{F}$ of each time series with parameter $\boldsymbol{\beta} = \{\theta_h, \mathbf{w}_{\mu}, \mathbf{w}_{\sigma}\}$, we can calculate their correlation in terms of Eq. (3). Specifically, we first compute the rank of each variable in their sequence as $\tilde{q}_{i,t}$, transform each variable based on the learned marginal distribution and inverse CDF of a standard normal distribution Φ^{-1}, where the transformed i-th time series denoted as:

$$\mathbf{u}_i = [\Phi^{-1}(\hat{F}_i(\tilde{q}_{i,1}; \beta_i)), ..., \Phi^{-1}(\hat{F}_i(\tilde{q}_{i,t\tau}; \beta_i)), ...] \tag{6}$$

Then, the loss for learning covariance matrix $\mathbf{R}$ in Eq. (3) can be written as the MLE form. As the estimated CDF is differentiable, the loss can be learned by stochastic gradient decent-based method:

$$\mathcal{L}_C = -\sum_{i=0}^{m} \left\{ \log \Phi_{\mathbf{R}}(\mathbf{u}_0, ..., \mathbf{u}_m) + \sum_t \log \hat{f}_i(\tilde{q}_{i,t}) \right\} \tag{7}$$

$\hat{f}_i(\cdot)$ is the probability density function (PDF) of $\hat{F}_i$. During training, the function $\Phi_{\mathbf{R}}(\cdot)$ requires to compute the inverse of $\mathbf{R}$, which would cause the numerical instability issue as it may be initialized in an ill-conditioned state. Owning to the Cholesky decomposition, the coefficient matrix can be decomposed as $\mathbf{R} = \mathbf{L}\mathbf{L}^{\top}$, $\mathbf{L}$ is the Cholesky lower-triangle matrix. Here we use an empirical guardrail (Wen and Torkkola 2019) to enforce a stable parameterization of $\mathbf{R}$: obtain the diagonal and off-diagonal elements of a raw $\mathbf{L}$ separately; then apply a sigmoid transform on diagonal elements and plus vector one to make sure values in diagonal are greater than 1; put a $\tanh(\cdot)$ activation on off-diagnonal; divide each row of the raw $\mathbf{L}$ by the raw l-2 norm; finally, the learned correlation matrix $\mathbf{L}\mathbf{L}^{\top}$ is stable.

By optimizing $\mathcal{L}_C$, the coefficients $\mathbf{R}$ among macro-factors and specific stock sector is learned. However, how both of the macroeconomic factors and the stock sector play a role on a specific stock is still unknown. For this reason, we design a gating function g_{gat} to integrate various factors and form the specific macro-sector context, which consists of a linear layer and a sigmoid function:

$$\boldsymbol{\alpha} = g_{gat}(\mathbf{R}) = Sigmoid(\mathbf{w}_R \mathbf{R} + \mathbf{b}_R) \tag{8}$$

Here $\mathbf{w}_R, \mathbf{b}_R, \boldsymbol{\alpha} \in \mathbb{R}^{1 \times (m+1)}$ have the same dimension. The specific macro-sector context embedding at time t denoted as $\mathbf{e}^q_t$, is derived from state embedding $\mathbf{h}^{\tau}_{i,t}$.

$$\mathbf{e}^q_t = \sum_{i=0}^{m} \alpha_i \mathbf{h}^{\tau}_{i,t} \tag{9}$$

Please note that $\boldsymbol{\alpha}$ depends on coefficients but is optimized and back-propagated mainly from stock-level information, hence we adopt the gating function to integrate the context instead of the direct coefficient matrix $\mathbf{R}$.

Stock Representation Learning

Pipeline For embedding robust stock representations, we connect the relevant macro-sector context $\mathbf{e}_t^q$ to future stock states $\mathbf{z}_{t+1}$, which macro-sector-micro framework forms. As in Figure 4, to capture such interaction and further train the encoder for stock representation, by shuffling future states and identify them as positive and negative samples, it could be transformed into predicting whether there is a connection between them. Since it is trained on observation data, this unsupervised architecture can express the stock embedding more accurately on the updated encoder g_{enc}.

Our primary goal is to learn a high-level stock representation from a macro and micro (stock)-coupled environment with low-level noises. Using labeled data for back-propagation may introduce some noise because labels contain less information, instead we adopt an unsupervised way to learn stock embedding. As explained in the Introduction, as the impact of macro-factors is reflected on future stock state, there is a connection between the historical context information and future stock state, and more contextual information is beneficial for time series forecasting (Wiskott and Sejnowski 2002), we thus calculate the mutual information $I(\mathbf{x};\mathbf{e})$ between future stock state $\mathbf{x}$ and present context information $\mathbf{e}$ based on CPC method as below to model the relationship between context information and future state:

$$I(\mathbf{x};\mathbf{e}) = \sum_{\mathbf{x},\mathbf{e}} p(\mathbf{x},\mathbf{e}) \log \frac{p(\mathbf{x}|\mathbf{e})}{p(\mathbf{x})} \tag{10}$$

The context information in the CPC model is also called 'slow feature' (e.g., phonemes and intonation in speech, or the story line in books). In our task, the 'slow feature' is incarnated as not only micro (stock context) but also macro and sector (macro-sector context) perspectives. By maximizing the mutual information $I(\mathbf{x};\mathbf{e})$, the impact of the current context on future specific stock state is well simulated. Specifically, the current macro-sector-micro context is formulated by an MLP after concatenating them together:

$$\mathbf{e}_t = \text{MLP}([\mathbf{e}_t^q, \mathbf{e}_t^x]) \tag{11}$$

Since the macro-sector context is varied when refer to different stock sector, we just focus on the specific one corresponding with the target stock. Here t is the demarcation point between past and future. It is randomly generated so that it can express historical information at different time windows, which also makes the model more general.

As shown in Figure 4, to learn the micro-level context $\mathbf{e}_t^x$, we first adopt a nonlinear encoder g_{enc} to map the input sequence of observed features $\mathbf{x}_t$ to a sequence of latent representations $\mathbf{z}_t = g_{enc}(\mathbf{x}_t)$. Next, an auto-regressive model (e.g., the GRUs method) g_{ar} summarizes all $\mathbf{z}_{\leq t}$ in the latent space to produce a stock context representation $\mathbf{e}_t^x = g_{ar}(\mathbf{z}_{\leq t})$, where $\mathbf{e}_t^x \in \mathbb{R}^{1\times d_x}$.

Similar to the CPC method, we do not predict future observations $\mathbf{x}_{t+k}$ directly with a generative model $p_k(\mathbf{x}_{t+k}|\mathbf{e}_t)$, instead, we model a density ratio which preserves the mutual information between $\mathbf{x}_{t+k}$ and $\mathbf{e}_t$ as:

$$f_k(\mathbf{x}_{t+k}, \mathbf{e}_t) \propto \frac{p(\mathbf{x}_{t+k}|\mathbf{e}_t)}{p(\mathbf{x}_{t+k})} \tag{12}$$

where $\propto$ means 'proportional to'. We simply adopt a log-bilinear model as density ratio function f_k:

$$f_k(\mathbf{x}_{t+k}, \mathbf{e}_t) = \exp\left(\mathbf{z}_{t+k}^\top \mathbf{W}_k \mathbf{e}_t\right) \tag{13}$$

The linear transformation $\mathbf{W}_k\mathbf{e}_t$ can be used for the prediction with a different $\mathbf{W}_k$ for every time step k. (Oord, Li, and Vinyals 2018) recommends that either of $\mathbf{z}_t$ and $\mathbf{e}_t$ could be used as the representation for downstream task. In our experiments, we prefer to use the learned encoder g_{enc} to embed stocks for stock movement prediction.

In simple terms, the Eq. (13) is to calculate the probability of connection between present context $\mathbf{e}_t$ and 'proposal' future state $\mathbf{z}_{t+k}$ exists. Here future sample $\mathbf{x}_{t+k}$ is shuffled and identified as a positive or negative sample. For example, if $\mathbf{x}_{t+k}$ is not come from the same stock as previous state $\mathbf{x}_t$, then it is labeled as a negative sample (the dashed green one in Figure 4), otherwise as positive sample (red one). Hence, $f_k(\cdot)$ calculates the probability that decides whether the future state at $t+k$ step is the same as the previous time series states. In this way, it can distinguish a future stock state that is inconsistent with the previous context. In other words, it can find the internal connection of the stock itself.

The loss function in CPC is based on noise contrastive estimation (NCE) technique (Gutmann and Hyvärinen 2010). Given a batch stocks $\mathbf{X} = \{\mathbf{X}_1, ..., \mathbf{X}_n\}$ and each stock feature sequence $\mathbf{X}_j = [\mathbf{x}_{j,1}, ..., \mathbf{x}_{j,T}]$. These n random samples contain positive samples from $p(\mathbf{x}_{t+k}|\mathbf{e}_t)$ and negative samples from the 'proposal' distribution $p(\mathbf{x}_{t+k})$ (which shuffled in the same batch), the loss function is:

$$\mathcal{L}_N = -\mathbb{E}_{\mathbf{X}}\left[\log \frac{f_k(\mathbf{x}_{t+k}, \mathbf{e}_t)}{\sum_{x_j\in\mathbf{X}} f_k(\mathbf{x}_j, \mathbf{e}_t)}\right] \tag{14}$$

Since the parameters in the first part also feed forward to stock representation learning, by integrating with the loss of copula-based model in Eq. (7), our total objective function for stock representation learning is to minimize:

$$\mathcal{L} = \frac{1}{2\gamma_1^2}\mathcal{L}_C + \frac{1}{2\gamma_2^2}\mathcal{L}_N + \log\gamma_1\gamma_2 \tag{15}$$

here γ_1 and γ_2 are parameters used to balance two losses (Kendall, Gal, and Cipolla 2018), the final loss could be learned based on SGD-based algorithms.

Prediction

For predicting stock movement, we first apply the encoder to map stock features into a generalized embedding space, then the decoder part could be any prediction model, e.g., LSTM, Attention-based LSTM, etc. In our experiment, we simply adopt LSTM as prediction model for verifying the general capability of the above stock representation.

Experiments

Experimental Settings

Datasets. We evaluate our model on two benchmarks for stock movement prediction: **ACL18** (Xu and Cohen 2018) and **KDD17** (Zhang, Aggarwal, and Qi 2017), the macroeconomic variables are from FRED (Federal Reserve Economic Data)[1]. **ACL18** contains 88 high-trade-volume-

[1]https://fred.stlouisfed.org/

stocks in NASDAQ, the features include stock prices and aligned Twitter text from 2014 to 2016. **KDD17** contains 50 stocks on yahoo.com from 2007 to 2016. As these two datasets are originally collected for stock price prediction rather than movements, we process the dataset similar to the original way [2] but not list here for brevity.

Refer to (Chen 2009), the macroeconomic variables in our experiment contain the 3-Month Treasury Bill Rate (TB3-daily, TB3W-weekly and TB3M-monthly), the 5-Year Treasury Constant Maturity Rate (DGS5-daily), the 10-Year Treasury Constant Maturity Rate (DGS10-daily), money stocks (M1 and M2 in weekly), aggregate output (industrial production in monthly, i.e., INDPRO), unemployment rates (UNRATE-monthly), nominal effective exchange rates (ER-monthly), federal funds rates (FFR-daily), the federal government debts (FGDS-quarterly), and inflation rates (consumer prices annually, i.e., CPI).

Baselines. We choose the following methods for comparison, the detailed settings for implementing ALSTM and Adv-ALSTM are the same as in (Feng et al. 2019), while the results of StockNet is copied from the origin paper.

- **StockNet** (Xu and Cohen 2018) applies a Variational Autoencoder (VAE) architecture to encode the stock input (includes historical prices and tweets) and decode stock movements with designed temporal attention.
- **LSTM** (Hochreiter and Schmidhuber 1997) is a widely used neural network for sequential prediction task, we implement it with an LSTM module and additional two linear layers and a sigmoid activation function for output. Noted that it is adopted as prediction in our model.
- **ALSTM** unites attention mechanism in LSTM-based recurrent neural network for consider information from different time-steps, here we follow the detailed setting in (Feng et al. 2019).
- **Adv-ALSTM** (Feng et al. 2019) introduces adversarial training into ALSTM for improving the generalization of a prediction model by adding adversarial perturbation of features on clean samples.
- **CPC** integrates the original CPC framework and our LSTM prediction part, i.e., excluding the copula part, it is optimized by the $\mathcal{L}_N$ loss and just involves stock context $\mathbf{e}_t^x$ as the present context.

Evaluation Metrics. Following the previous work for stock movement prediction (Ding et al. 2016), we adopt the standard measure of accuracy (Acc.) and Matthews Correlation Coefficient (MCC) as evaluation metrics, the higher of these two metrics the better of performance.

Parameter Settings. Our Co-CPC model is implemented with Pytorch, optimizes Eq. (15) by Adam and optimizes the prediction part by Adagrad with a batch size of 32. We use a 20-day lag window for sample construction to learn historical context. The embedding size of stock context and macro context are set at 64 for controlling memory costs and make training feasible in a single GPU. In the second part, we simply apply the GRU module in Pytorch as the auto-regression, and the weight is initialized with kaiming normal (He et al. 2018). As for the encoder, it consists of a linear layer with a ReLU activation function. Because of page limitations, more specific implementations can refer to our code page [3].

[2]https://github.com/yumoxu/stocknet-dataset

Method	ACL18		KDD17	
	Acc.	MCC	Acc.	MCC
StockNet	54.96	0.00165	51.93	0.0335
LSTM	53.18	0.0674	51.62	0.0183
ALSTM	54.90	0.1403	51.94	0.0261
Adv-ALSTM	57.20	0.1483	53.05	0.0523
CPC	58.14	0.1631	54.47	0.0746
Co-CPC	**58.90**	**0.1771**	**58.81**	**0.1643**
StockNet	58.23	0.0808	-	-
LSTM	56.82	0.1375	-	-
CPC	59.11	0.1817	-	-
Co-CPC	**61.62**	**0.2316**	-	-

Table 1: Performance comparison on two datasets.

Experimental Results

Short-term Prediction Performance. To demonstrate the Co-CPC effectiveness, we compare it with the above baselines in Table 1. Since the stock feature in ACL18 dataset contains additional tweet information, we compare the result in two aspects, which the upper part only consider stock prices, while the lower part both consider prices and tweet features. From the table, we have following observations:

- Co-CPC achieves the best results in all the cases specifically compared with the CPC, which explains that the stock price fluctuations is partially due to the released macro factors. What's more, it indicates the coupled three level relationship (macro-sector-micro) can be well captured by our design.
- Compare the results upper and lower parts in terms of price only and extra tweet information added situation, the improvement illustrates the tweet text contains much more potential factors for stock movement. In addition, although the processing way about tweet is the same as StockNet, Co-CPC still outperforms them. This may be because StockNet and LSTM use label information for back-propagation in learning stock hidden representation, while Co-CPC does not use any labels in back-propagation but predict the true or false of stock states at future moment which makes the generalization of the stock representation better.
- Among the baselines, ALSTM outperforms LSTM reflecting the impact of attention mechanism. Although our prediction is based on LSTM, the good performance verifies the superiority of our model. Besides, the Adv-ALSTM outperforms ALSTM because it can better capture the existence of uncertainty in stock prices. To compare with Adv-ALSTM, the better performance of Co-CPC reflects that it further reduces the uncertainty by cou-

[3]https://github.com/goiter/CoCPC

Method	ACL18		KDD17	
	Acc.	MCC	Acc.	MCC
CPC	59.11	0.1817	54.47	0.0746
Co-CPC(monthly)	59.54	0.1892	56.94	0.1201
Co-CPC(weekly)	59.76	0.1960	57.02	0.1221
Co-CPC(daily)	60.03	0.2006	56.63	0.1142

Table 2: Performance comparison when consider macro-factors with various time intervals. The results of ACL18 dataset here are based on both prices and tweets features.

pling macro-factors, sector features and stock prices, as well as learning the stock representation.

Macro-variable Impact Analysis. We now investigate the impacts of macro-factors on stock movement prediction from two perspectives. To be more specific, we implement some variances of our model as shown in Table 2. The results of CPC reflect the situation without considering macro-factors, the other methods are Co-CPC with only considering macro-factors of specific time intervals. The gap between CPC and the other three methods reflects the role of macro-factors and the efficiency of our macro-sector-micro coupled design for stock movement prediction. Comparing with two datasets, it seems that stocks in NASDAQ (ACL18 dataset) are more vulnerable to daily macro factors (e.g., Treasury Bill Rate), while stocks from the yahoo website (KDD17 dataset) are more likely to be affected by weekly macro factors (e.g., M1 and M2 money stock).

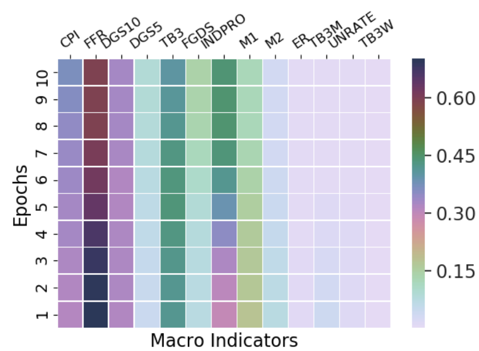
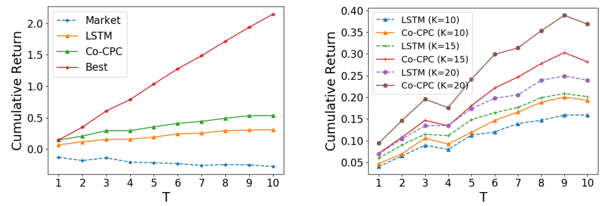


Figure 5: The impact degree of macro indicators.

Further, we record the weights (Eq. (8)) in the 10 epochs of stock representation learning on ACL18 to see how distinct macro-factors affect the training. By normalizing the gating value α, we draw the varied impact degree in Figure 5. From the comparison of each column, the top five most influential factors include the daily (e.g., FFR, DGS10), monthly (e.g., INDPRO) and annually (e.g., CPI) macro-factors. It shows that stock movement is not only related to recently released indicators but also to indicators with a large span. From the comparison of each row, the weights change under different epochs are relatively slight, reflecting the stability of the learned macro-sector context in the face of various unknown data.



(a) 1-th strategy (b) 2-th strategy

Figure 6: Cumulative return simulation on two strategies.

Long-term Prediction Performance. In addition, to validate our learned stock representation can also be benefit for long-term prediction, we conduct an experiment on forecasting the stock movement in the next 10 days and simulate its cumulative return (Liu et al. 2020) for a more intuitive comparison. Two kinds of investment strategies are applied for validating the performance more comprehensively.

Simply, the first investment strategy is to select all the stocks with positive return per day. As shown in Figure 6(a), the market result here is to hold all the stocks for indicating the overall market trend, while the best result is based on the true positive return which represents the perfect situation. Since the baselines in Table 1 adopt specific predictor and just focus on next moment prediction, here we just compare with limited methods. Although our Co-CPC method could not achieve the best revenue, at least it beats others when the entire economic market downturn (the blue dash line).

The second investment strategy is based on the estimated scores, whose values equal to the probability to have a rising trend minus the probability to have a declining trend. Then the straightforward portfolio contains top-K stocks with the highest scores and positive returns in the next trading day. Generally, investors always choose multiple stocks to avoid risks, so we compare the results with different K in Figure 6(b). It can be seen that no matter which K sets, our Co-CPC performs better than LSTM. Besides, the more stocks chose, the more revenue Co-CPC obtains than LSTM.

Conclusion

On the purpose of alleviating data and modeling uncertainty in stock movement prediction, we propose a Co-CPC model to represent stocks more robustly. Specifically, under the designed macro-sector-micro framework, heterogeneous macroeconomic variables are reasonably considered into the coupling relations between macro-level and sector-level variables for easing the data uncertainty. Besides, in order to learn a generalized stock representation for reducing the modeling uncertainty, the macro-sector contexts are integrated with the micro-stock context and a self-supervised task captures their couplings. The conducted experiments validate the efficiency of our model in both short-term and long-term stock movement prediction. Though we mainly evaluate the performance on stock movement prediction, our model is general and can be extended for other time series forecasting tasks while taking complicated relations (i.e., cross-temporal couplings) between multiple time series.

Acknowledgements

This work is supported in part by Australian Research Council Discovery Grant (DP190101079), ARC Future Fellowship Grant (FT190100734).

References

Adam, A. M.; and Tweneboah, G. 2008. Macroeconomic factors and stock market movement: Evidence from Ghana. *Available at SSRN 1289842* .

Bai, S.; Kolter, J. Z.; and Koltun, V. 2018. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271* .

Beber, A.; and Brandt, M. W. 2008. Resolving macroeconomic uncertainty in stock and bond markets. *Review of Finance* 13(1): 1–45.

Cao, L. 2015. Coupling learning of complex interactions. *Inf. Process. Manage.* 51(2): 167–186.

Cao, L. 2020. AI in Finance: A Review Http://dx.doi.org/10.2139/ssrn.3647625.

Cao, L.; Ou, Y.; and Yu, P. S. 2012. Coupled Behavior Analysis with Applications. *IEEE Trans. Knowl. Data Eng.* 24(8): 1378–1392.

Cao, W.; Hu, L.; and Cao, L. 2015. Deep modeling complex couplings within financial markets. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.

Chatigny, P.; Chen, R.; Patenaude, J.-M.; and Wang, S. 2018. A Variable-Order Regime Switching Model to Identify Significant Patterns in Financial Markets. In *2018 IEEE International Conference on Data Mining (ICDM)*, 887–892. IEEE.

Chen, S.-S. 2009. Predicting the bear stock market: Macroeconomic variables as leading indicators. *Journal of Banking & Finance* 33(2): 211–223.

Choroś, B.; Ibragimov, R.; and Permiakova, E. 2010. Copula estimation. In *Copula theory and its applications*, 77–91. Springer.

Ding, X.; Zhang, Y.; Liu, T.; and Duan, J. 2016. Knowledge-driven event embedding for stock prediction. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, 2133–2142.

Feng, F.; Chen, H.; He, X.; Ding, J.; Sun, M.; and Chua, T.-S. 2019. Enhancing stock movement prediction with adversarial training. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, 5843–5849. IJCAI.

Gao, T.; Li, X.; Chai, Y.; and Tang, Y. 2016. Deep learning with stock indicators and two-dimensional principal component analysis for closing price prediction system. In *2016 7th IEEE International Conference on Software Engineering and Service Science (ICSESS)*, 166–169. IEEE.

Gutmann, M.; and Hyvärinen, A. 2010. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, 297–304.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2018. Delving deep into rectifiers: surpassing human-level performance on ImageNet classification. ArXiv e-prints 2015.

Hochreiter, S.; and Schmidhuber, J. 1997. Long short-term memory. *Neural computation* 9(8): 1735–1780.

Ibrahim, M. 1999. Macroeconomic variables and stock prices in Malaysia: An empirical analysis. *Asian Economic Journal* 13(2): 219–231.

Kendall, A.; Gal, Y.; and Cipolla, R. 2018. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7482–7491.

Leontjeva, A.; and Kuzovkin, I. 2016. Combining Static and Dynamic Features for Multivariate Sequence Classification. In *DSAA'2016*, 21–30. IEEE.

Lin, T.; Guo, T.; and Aberer, K. 2017. Hybrid neural networks for learning the trend in time series. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, 2273–2279. IJCAI.

Liu, Y.; Liu, Q.; Zhao, H.; Pan, Z.; and Liu, C. 2020. Adaptive Quantitative Trading: An Imitative Deep Reinforcement Learning Approach. AAAI.

Nti, K. O.; Adekoya, A.; and Weyori, B. 2019. Random Forest Based Feature Selection of Macroeconomic Variables for Stock Market Prediction. *American Journal of Applied Sciences* 16(7): 200–212.

Oord, A. v. d.; Li, Y.; and Vinyals, O. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748* .

Qin, Y.; Song, D.; Cheng, H.; Cheng, W.; Jiang, G.; and Cottrell, G. W. 2017. A dual-stage attention-based recurrent neural network for time series prediction. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, 2627–2633. IJCAI.

Salinas, D.; Bohlke-Schneider, M.; Callot, L.; Medico, R.; and Gasthaus, J. 2019. High-dimensional multivariate forecasting with low-rank Gaussian Copula Processes. In *Advances in Neural Information Processing Systems*, 6824–6834.

Sklar, A. 1973. Random variables, joint distribution functions, and copulas. *Kybernetika* 9(6): 449–460.

Stefani, J. D.; Borgne, Y. L.; Caelen, O.; Hattab, D.; and Bontempi, G. 2019. Batch and incremental dynamic factor machine learning for multivariate and multi-step-ahead forecasting. *Int. J. Data Sci. Anal.* 7(4): 311–329.

Sun, T.; Wang, J.; Zhang, P.; Cao, Y.; Liu, B.; and Wang, D. 2017. Predicting stock price returns using microblog sentiment for chinese stock market. In *2017 3rd International Conference on Big Data Computing and Communications (BIGCOM)*, 87–96. IEEE.

Vo, X. V.; and Phan, D. B. A. 2019. Herd behavior and idiosyncratic volatility in a frontier market. *Pacific-Basin Finance Journal* 53: 321–330.

Wen, R.; and Torkkola, K. 2019. Deep generative quantile-copula models for probabilistic forecasting. *arXiv preprint arXiv:1907.10697* .

Wiskott, L.; and Sejnowski, T. J. 2002. Slow feature analysis: Unsupervised learning of invariances. *Neural computation* 14(4): 715–770.

Xu, Y.; and Cohen, S. B. 2018. Stock movement prediction from tweets and historical prices. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1970–1979.

Zhang, L.; Aggarwal, C.; and Qi, G.-J. 2017. Stock price prediction via discovering multi-frequency trading patterns. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, 2141–2149. ACM.

Zhang, X.; Zhang, Y.; Wang, S.; Yao, Y.; Fang, B.; and Philip, S. Y. 2018. Improving stock market prediction via heterogeneous information fusion. *Knowledge-Based Systems* 143: 236–247.