Arbitrage-free yield curve and bond price forecasting by deep neural networks

Cody Hyndman
cody.hyndman@concordia.ca
Department of Mathematics and Statistics
(with Xiang Gao)

September 29, 2021

Background

- The objective of many machine learning models used in mathematical finance is to predict asset prices by learning functions depending on stochastic inputs.
- In general, there is no guarantee that these stochastic factor models are consistent with no-arbitrage conditions.
- In Kratsios and Hyndman [5] we introduced a regularization approach to model selection within a generalized HJM framework.
- We consider a general framework for learning the arbitrage-free factor model that is most similar to a factor model within a prespecified class of alternative factor models.
- The incorporation of an arbitrage-penalty term allows various machine learning methods to be directly and coherently integrated into mathematical finance applications.

Background

- The inuition of [5] is to consider the following regularization problem

$$\underset{\phi \in \mathcal{H}}{\arg\min}\ \ell\left(\varphi - \phi\right) + AF^{\lambda}(\phi); \tag{0.1}$$

  where ℓ is a loss function, AF^{λ} is an arbitrage penalty, and $\mathcal{H}$ is a class of factor models.
- For example, consider the Nelson-Siegel family as part of a larger class of affine term-structure models, in which, at any given time, the forward-rate curve is described in terms of a set of market factors as

$$\varphi(t, \beta, u) \triangleq \varphi_0(u - t) + \sum_{i=1}^{d} \beta^i \varphi_i(u - t) \tag{0.2}$$

- The Nelson-Siegel model is typically not arbitrage-free therefore we would like to learn the closest arbitrage-free factor model, driven by the same stochastic factors.

Background

- In [5] we considered various predictive models for the term structure in this framework and used a data set of German bond data from 2010 to 2014.
- In this presentation we build on [5] to consider a more robust modelling and predictive approach to US treasuries and corporate bond issues from 2017 to 2019.
- Mathematical and implementation details can be found in:

  Gao, Xiang. *Stochastic control, numerical methods, and machine learning in finance and insurance.* PhD Thesis, Concordia University, May 2021.

  `https://spectrum.library.concordia.ca/988412/` or the forthcoming arXiv preprint.

Background

- Dynamic Nelson-Siegel term structure: Diebold and Li [4]

$$y(t,\tau) = L_t + S_t\left(\frac{1-e^{-\lambda\tau}}{\lambda\tau}\right) + C_t\left(\frac{1-e^{-\lambda\tau}}{\lambda\tau} - e^{\lambda\tau}\right).$$

- Arbitrage-free Nelson-Siegel in autoregression model: Ang and Piazzesi [1]

$$X_t = \mu + \Phi X_{t-1} + u_t.$$

  where $X_t = (L_t, S_t, C_t)$.
- Affine arbitrage-free term structure: Christensen et al. [3]

$$y(t,\tau) = B(t,T)X_t + C(t,T).$$

- Our interest:
  - Regularization: Arbitrage-free restriction.
  - Machine learning: Predicting bond prices.

Forward rate model and factor model

- Forward rate: HJM under the risk-neutral measure $\mathbb{Q}$, for $\tau = T - t$

$$df\,(t,\tau) = \mu\,(t,\tau)\,dt + \sum \eta_i\,(t,\tau)\,dW_i(t). \tag{0.3}$$

- Affine model: Assume $f(t,\tau)$ is separable for t and τ

$$f(t,\tau) = \beta_\tau X_t,$$

for a loading parameter $\beta_\tau \in \mathbb{R}^{1\times d}$ and a risk process $X_t \in \mathbb{R}^{d\times 1}$.

- Factor model: Risk factor X_t as Vašíček process

$$dX_t = \kappa_t\,(\theta_t - X_t)\,dt + \sigma_t dW_t. \tag{0.4}$$

X_t has explicit form

$$X_t = e^{-\int_0^t \kappa_u du} X_0 + \int_0^t e^{-\int_u^t \kappa_v dv} \kappa_u \theta_u du + \int_0^t e^{-\int_u^t \kappa_v dv} \sigma_u dW_u.$$

- Mean-reverting forward rate:

$$df(t,\tau) = \bar{\kappa}(t,\tau)\left(\bar{\theta}(t,\tau) - X_t\right)dt + \beta_\tau \sigma_t dW_t,$$

where

$$\bar{\kappa}(t,\tau) = \left(\beta_\tau \kappa_t + \frac{d\beta_\tau}{d\tau}\right),\ \bar{\theta}(t,\tau) = \left(\beta_\tau \kappa_t + \frac{d\beta_\tau}{d\tau}\right)^{-1} \beta_\tau \kappa_t \theta_t.$$

Dynamic Nelson-Siegel

- Loading parameter: β_τ is a three-dimensional vector basis for some $\lambda \in \mathbb{R}^+$
$$\beta_\tau = \left(1,\ e^{-\lambda\tau},\ \lambda\tau e^{-\lambda\tau}\right).$$
- Nelson-Siegel (NS) space $\mathcal{NS}(\tau)$ is spanned by the basis β_τ
$$\mathcal{NS}(\tau) = \text{Span}\left\{ \left(1,\ e^{-\lambda\tau},\ \lambda\tau e^{-\lambda\tau}\right) \middle| \text{ for some } \lambda \in \mathbb{R}^+ \right\}.$$
- Forward rate: $f(t,\tau) \in \mathcal{NS}(\tau)$ for $X_t = (X_1(t), X_2(t), X_3(t))^T \in \mathbb{R}^{3\times 1}$
$$f(t,\tau) = \beta_\tau X_t = X_1(t) + e^{-\lambda\tau} X_2(t) + \lambda\tau e^{-\lambda\tau} X_3(t).$$
- Yield curve: For $B_\tau = \int_0^\tau \beta_u du$
$$y(t,\tau) = \frac{B_\tau}{\tau} X_t = X_1(t) + X_2(t)\left(\frac{1-e^{-\lambda\tau}}{\lambda\tau}\right) + X_3(t)\left(\frac{1-e^{-\lambda\tau}}{\lambda\tau} - e^{\lambda\tau}\right).$$
- Completeness of NS space: As shown by Björk and Svensson [2], as long as the drift and volatility of the forward rate process lies in $\mathcal{NS}(\tau)$ whose tangent space is itself then the forward rate process will evolve in $\mathcal{NS}(\tau)$.

Arbitrage-free forward rate model

Theorem 0.1 (Arbitrage-free forward rate)

Suppose the forward rate model has an affine structure give by $f(t,\tau) = \beta_\tau X_t$*, and a mean-reverting state variable defined by (0.4)*

$$dX_t = \kappa_t\left(\theta_t - X_t\right)dt + \sigma_t dW_t,$$

where $\beta_\tau \in \mathbb{R}^{1\times d}$, $X_t \in \mathbb{R}^{d\times 1}$, *and*

$$\begin{aligned}
\kappa_t\left(X_t\right) :&\mathbb{R}^{d\times 1} \to \mathbb{R}^{d\times d},\\
\theta_t\left(X_t\right) :&\mathbb{R}^{d\times 1} \to \mathbb{R}^{d\times 1},\\
\sigma_t\left(X_t\right) :&\mathbb{R}^{d\times 1} \to \mathbb{R}^{d\times d}.
\end{aligned}$$

If the excess return is 0 *for all* $t \geq 0$ *and* $\tau \geq 0$*.*

$$\Lambda(t,\tau) = \frac{1}{2}B_\tau \Sigma_t B_\tau^T - B_\tau\kappa_t\left(\theta_t - X_t\right) + \left(\beta_\tau - \beta_0\right)X_t = 0,$$

then $f(t,\tau)$ *is arbitrage-free forward rate model under risk-neutral measure* $\mathbb{Q}$*.*

Bond data

- Corporate bond market (TRACE) and U.S.Treasuries from 2017 to 2019 (daily closed)
- Total data: 60 $\sim$ 84 daily bonds.
  i. Fixed coupon bond
  ii. Non-callable bond
  iii. Nonconvertible bond
  iv. Yield-to-maturity $\leq$ 700 basis points
  v. The difference between yield-to-maturity and coupon rate $\leq$ 500 basis points
  vi. Remaining time to the maturity $\geq$ 3 months and $\leq$ 30 years from the trade date
- Four features: Clean price (daily closed), coupon rate, coupon frequency, and the remaining time to maturity (tenor).

Data selection

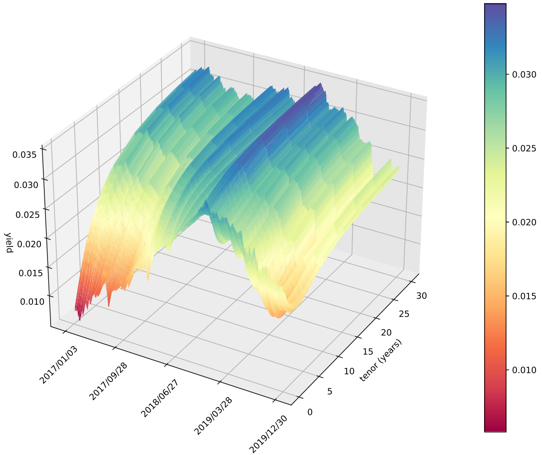
- First regression:
  - Common data (in proportion): 67 daily observations from the total data
    - Short-term tenor 0$\sim$2 years: 20%, 14 bonds
    - Mid-term tenor 2$\sim$15 years: 67%, 45 bonds
    - Long-term tenor 15$\sim$30 years: 13%, 9 bonds
  - Dropout: Trading date with too few observation (around 1% to 3%).
  - Objective: Determine the optimal decay parameter $\lambda = 0.4488779759$.
- Second regression:
  - Total data: Fit daily data with fixed $\lambda = 0.4488779759$.
  - Sequential fitting: Use the optimal X_{t-1} as initials to obtain the optimal X_t.
  - Objective: Extract optimal state variables X_t, for $t = 1, 2, \cdots$.
- Price of coupon bond given by the arbitrage-free Nelson-Siegel model

$$\hat{Y}(t,\tau) = \sum_{i=1}^{m} c_i e^{-\tau_i y(\tau_i)} = \sum_{i=1}^{m} c_i e^{-B_{\tau_i} X_t}.$$

- Minimizing the MSE

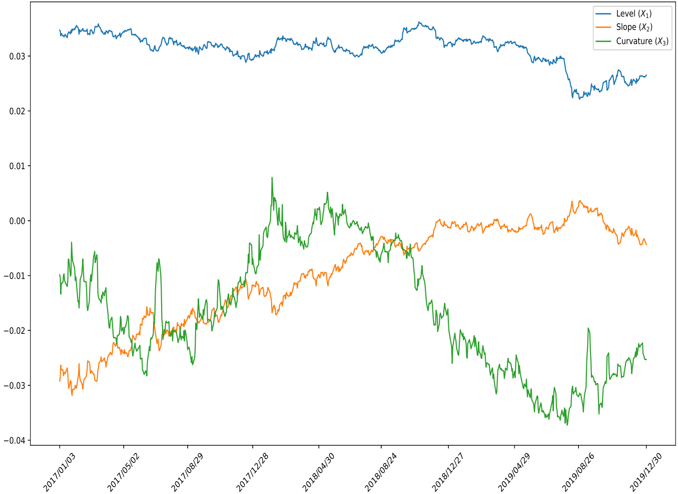
$$X_t = \underset{X_t \in \mathbb{R}^d}{\arg\min} \frac{1}{n} \sum_{i=1}^{n} \left| Y(t,\tau_i) - \hat{Y}(t,\tau_i) \right|^2.$$

Figure 1: US Treasuries yield surface



Dynamic NS State Variables

Figure 2: State Variables 2017 to 2019



Prediction framework with filters

- We introduce three filter-based sequential methods to estimate bond yields and bond prices given the parameters $(\kappa_t, \theta_t, \sigma_t)$.
- We then we introduce the estimation of parameters $(\kappa_t, \theta_t, \sigma_t)$ by deep neural networks.
- The Kalman filter and the extended Kalman filter have similar structure but one deals with yields prediction and the other deals with bond price prediction.
- We begin by introducing the Kalman filter in detail then extend the prior and post information updating method to the extended Kalman filter and the particle filter.

Prediction framework with filters

- Predict yields $\hat{y}_t$ using predicted state variable $\hat{X}_t$

$$\hat{y}(t,\tau) = \frac{B_\tau}{\tau}\hat{X}_t + \epsilon_t,$$

where $\mathbb{E}\left[\epsilon_t\right] = 0$ and $\mathrm{Var}\left[\epsilon_t\right] = U_t$.

- Predict bond prices $\hat{Y}_t$

$$\hat{Y}(t,\tau) = \sum_{i=1}^{m} c_i e^{-\tau_i y(\tau_i)} + \epsilon_t = \sum_{i=1}^{m} c_i e^{-B_{\tau_i}\hat{X}_t} + \epsilon_t. \tag{0.5}$$

- We assume κ_t, θ_t, and σ_t are invariant on each time interval $t \in [t_k, t_{k+1})$ and $\Delta t = t_{k+1} - t_k$ are the same for all k,

$$\mathbb{E}\left[X_{k+1}|\mathcal{F}_k\right] = e^{-\kappa_k \Delta t}X_k + \left(I - e^{-\kappa_k \Delta t}\right)\theta_k, \tag{0.6}$$

$$\mathrm{Var}\left[X_{k+1}|\mathcal{F}_k\right] = \int_0^{\Delta t} e^{-\kappa_k u}\Sigma_k e^{-\kappa_k^T u}du. \tag{0.7}$$

Kalman Filter

- The prediction step

$$\hat{X}_{k|k-1} = A_{k-1}\hat{X}_{k-1|k-1} + D_{k-1},$$
$$\hat{P}_{k|k-1} = A_{k-1}\hat{P}_{k-1|k-1}A_{k-1}^T + Q_{k-1},$$

- The measurement step

$$\begin{aligned}
\hat{X}_{k|k} &= \hat{X}_{k|k-1} + K_k v_k, \\
\hat{P}_{k|k} &= \hat{P}_{k|k-1} - K_k M \hat{P}_{k|k-1}, \\
v_k &= y_k - M\hat{X}_{k|k-1}, \\
F_k &= M\hat{P}_{k|k-1}M^T + U_{k-1}, \\
K_k &= \hat{P}_{k|k-1}M^T F_k^{-1},
\end{aligned} \tag{0.8}$$

where

$$\begin{aligned}
A_t &= e^{-\kappa_t \Delta t}, \\
D_t &= \left(I - e^{-\kappa_t \Delta t}\right)\theta_t, \\
Q_t &= \mathrm{Var}\left[X_{t+1}|\mathcal{F}_t\right],
\end{aligned}$$

$$M = \begin{bmatrix} \frac{B_1(\tau_1)}{\tau_1}, & \frac{B_2(\tau_1)}{\tau_1}, & \frac{B_3(\tau_1)}{\tau_1} \\ \vdots & \vdots & \vdots \\ \frac{B_1(\tau_m)}{\tau_m}, & \frac{B_2(\tau_m)}{\tau_m}, & \frac{B_3(\tau_m)}{\tau_m} \end{bmatrix}.$$

Prediction framework: Extended Kalman Filter

- Non-linear model for prices of coupon bonds.

$$\hat{Y}(X_t, t) = \sum_{j}^{m} c_{\tau_j} e^{-B_{\tau_j} X_t} = C_\tau \exp\left(-B_\tau X_t\right),$$

where

$$\begin{aligned} C_\tau &= (c_{\tau_1}, c_{\tau_2}, \cdots, c_{\tau_m}) \in \mathbb{R}^{1\times m}, \\ B_\tau &= (B_{\tau_1}, B_{\tau_2}, \cdots, B_{\tau_m})^T \in \mathbb{R}^{m\times 3}. \end{aligned}$$

- The prediction step

$$\begin{aligned} \hat{X}_{k|k-1} &= A_k \hat{X}_{k-1|k-1} + D_k, \\ \hat{P}_{k|k-1} &= A_k \hat{P}_{k-1|k-1} A_k^T + Q_k, \end{aligned}$$

where the Hessian matrix M_k is

$$M_k = \frac{\partial \hat{Y}(X,t)}{\partial X}\Big|_{(X,t)=(\hat{X}_k, t_k)}$$

- The measurement process

$$\begin{aligned} \hat{X}_{k|k} &= \hat{X}_{k|k-1} + K_k v_k, \\ \hat{P}_{k|k} &= \hat{P}_{k|k-1} - K_k M_k \hat{P}_{k|k-1}, \\ v_k &= Y_k - \hat{Y}(\hat{X}_{k|k-1}, t_k), \\ F_k &= M_k \hat{P}_{k|k-1} M_k^T + U_k, \\ K_k &= \hat{P}_{k|k-1} M_k^T F_k^{-1}, \end{aligned}$$

Prediction framework: Particle filter

- State variables X_t as particles: Sample X_t from empirical distribution

$$X_t \sim X_t | Y_{t-1}.$$

- Multivariate generalized Gaussian distribution (MGGD): Control shape and kurtosis

$$Y_t \left| X_t \right. \sim \mathcal{M}\left(Y_t \left| \bar{Y}\left(X_t, t\right)\right.\right).$$

- Following the definition given by Pascal et al. [6], we define the density

$$q(x \left| \bar{x} \right.) = \left| U \right|^{-\frac{1}{2}} C_{p,n} \exp\left(- \frac{\left[\left(x - \bar{x}\right)^T U^{-1} \left(x - \bar{x}\right) \right]^p}{2m^p} \right).$$

- p is the shape ($p = 1.5$) and m is the scale ($m =$ number of x)
- $U \in \mathbb{R}^{n \times n}$ is the variance matrix and $C_{p,n}$ is a normalization coefficient

$$C_{p,n} = p\left(2^{\frac{1}{p}} \pi m\right)^{-\frac{n}{2}} \Gamma\left(\frac{n}{2}\right) \Big/ \Gamma\left(\frac{n}{2p}\right).$$

- $p = 0.5$, multivariate Laplace distribution; $p = 1$, multivariate Gaussian

Particle filter: Importance sampling and Resampling

- Difficult to sample from the posterior distribution $p(X_k \,|\, Y_{1:k})$
- Importance sampling: Sample from some prior distribution $q(X_k \,|\, Y_{1:k})$

$$\mathbb{E}^p \left[f(X_t) \,|\, Y_{1:t} \right] = \frac{\mathbb{E}^q \left[w_t(X_t) f(X_t) | Y_{1:t} \right]}{\mathbb{E}^q \left[w_t(X_t) | Y_{1:t} \right]},$$

where

$$w_t(X_t) = \frac{p(Y_{1:t}|X_t)p(X_t)}{q(X_t|Y_{1:t})}.$$

- Problems:
  - Degeneration (variance explodes)
  - Effectiveness of particles declines
- Resampling:
  - Apply unbiased distribution to update particles
  - The particle with big weight has more offspring

Particle filter: Sequential importance resampling (SIR) particle filter

At time $k = 0$, start with a common initial for all sequences

- Sample $X_0^{(i)}$ from the initial states

$$X_0^{(i)} = \hat{X}_0 + \hat{P}_0 W^{(i)},$$

  where $P_0 = \hat{P}_0 \hat{P}_0^T$ is the prior covariance matrix and $W^{(i)}$ is standard Gaussian random number.

- Updating weights by initial observations and resampling to obtain equally distributed particles $\{X_0^{(i)}, w_0^{(i)} = 1/N\}$

From time $k \geq 1$

- Importance sampling: Using the measurement and updating equations in EKF, we obtain the posterior particles along with the posterior covariance

$$\begin{aligned}\hat{X}^{(i)}_{k-1|k-1} &= X^{(i)}_{k-1} + K_{k-1} v^{(i)}_{k-1}, \\ P^{(i)}_{k-1|k-1} &= P^{(i)}_{k-1} - K_{k-1} M_{k-1} P^{(i)}_{k-1},\end{aligned}$$

then we sample particles from the posterior space

$$X^{(i)}_k = A_{k-1} \hat{X}^{(i)}_{k-1|k-1} + D_{k-1} + \sqrt{P^{(i)}_k} W^{(i)},$$

where

$$\begin{aligned}P^{(i)}_k &= A^T_{k-1} P^{(i)}_{k-1|k-1} A_{k-1} + Q_{k-1}, \\ v^i_{k-1} &= Y_{k-1} - \hat{Y}(t_{k-1},\ X^{(i)}_{k-1}), \\ F_{k-1} &= M_{k-1} P^{(i)}_{k|k-1} M^T_{k-1} + U_{k-1}, \\ K_{k-1} &= P^{(i)}_{k-1} M^T_{k-1} F^{-1}_{k-1}, \\ M_{k-1} &= \left.\frac{\partial \hat{Y}}{\partial X}\right|_{X = X^{(i)}_{k-1}}.\end{aligned}$$

- Update weights:

$$w_k^{(i)} = w_{k-1}^{(i)} \frac{p\left(Y_k \middle| X_k^{(i)}\right) p\left(X_k^{(i)} \middle| X_{k-1}^{(i)}\right)}{q\left(X_k^{(i)} \middle| x_{k-1}^{(i)}, Y_{1:k}\right)},$$

where

$$p\left(Y_k \middle| X_k^{(i)}\right) = \mathcal{M}\left(Y_k \middle| \hat{Y}(X_k^{(i)}), U_k\right),$$
$$p\left(X_k^{(i)} \middle| X_{k-1}^{(i)}\right) = \mathcal{N}\left(X_k^{(i)} \middle| g(X_{k-1}^{(i)}), Q_{k-1}\right),$$
$$q\left(X_k^{(i)} \middle| X_{k-1}^{(i)}, Y_{1:k}\right) = \mathcal{N}\left(X_k^{(i)} \middle| g(X_{k-1|k-1}^{(i)}), P_k^{(i)}\right).$$

Calculate normalized weights

$$\bar{w}_k^{(i)} = \frac{w_k^{(i)}}{\sum_{i=1}^N w_k^{(i)}}.$$

- Resampling: Obtain equally weighted sample $\left\{\frac{1}{N}, X_k^{(i)}, P_k^{(i)}\right\}$ from $\left\{\bar{w}_k^{(i)}, X_k^{(i)}, P_k^{(i)}\right\}$.
  - N equally spaced seeds $s_k^{(i)} = \frac{i-1+\tilde{s}_k}{N}$ for $\tilde{s}_k \sim U[0,1]$, $i = 1, \cdots, N$.
  - Offspring: number of seeds between two consecutive accumulative weights

$$o_k^{(i)} = \left| \left\{ s_k^{(j)} : \sum_{k=1}^{i-1} \bar{w}_k^{(n)} \le s_k^{(j)} \le \sum_{k=1}^{i} \bar{w}_k^{(n)} \right\} \right|,$$

which is the number of $s_k^{(j)}$ in $\left[\sum_{n=1}^{i-1} \bar{w}_k^{(n)}, \sum_{n=1}^{i} \bar{w}_k^{(n)}\right]$.

Dynamic parameterization by recurrent neural networks

- Filter-based RNN for prediction
- Model structure: Input layer, Residual layer, State layer, and Filter layer.
  - Model parameters:
    - Interpretable parameters: state parameters, state variables
    - Uninterpretable parameters: network parameters
  - Sequential model:
    - LSTM/GRU for sequential data
    - Convolution LSTM (CLSTM) for sequential image data
- In neural networks:
  - Learning for the state parameters $(\kappa_t, \theta_t, \sigma_t)$ from historical data

$$\kappa_t, \theta_t, \sigma_t = \mathcal{D}(Y_{t-T:t}, \vartheta).$$

  - Learning for the error parameter U_t from the prediction error $\left| Y_t - \hat{Y}_t \right|$
- In filter:
  - Update state variable X_t using $(\kappa_t, \theta_t, \sigma_t)$ and predict next state X_{t+1}
  - Predict yields or prices using predicted X_{t+1} and measure the error using the observation Y_{t+1}

Input layer and state layer

- Input layer: for yield input $y_t \in B \times T \times F$ and bond input $Y_t \in B \times T \times N \times F$

Two connected LSTM:

$$\begin{aligned}\left(c_t^{l_1}, h_t^{l_1}\right) &= L_1\left(y_t, \left(c_{t-1}^{l_1}, h_{t-1}^{l_1}\right)\right),\\ \left(c_t^{l_2}, h_t^{l_2}\right) &= L_2\left(c_t^{l_1}, \left(c_{t-1}^{l_2}, h_{t-1}^{l_2}\right)\right),\\ c_t^{l} &= c_t^{l_2}.\end{aligned}$$

Connected CLSTM and LSTM

$$\begin{aligned}Y_t \in \mathbb{R}^{N\times F} &:\to Y_t' \in \mathbb{R}^{N\times F\times 1},\\ \left(c_t^{l_c}, h_t^{l_c}\right) &= L_c\left(Y_t', \left(c_{t-1}^{l_c}, h_{t-1}^{l_c}\right)\right),\\ c_t^{l_c} \in \mathbb{R}^{H\times 1\times 1} &:\to c_t' \in \mathbb{R}^{1\times H},\\ \left(c_t^{l}, h_t^{l}\right) &= L\left(c_t', \left(c_{t-1}^{l}, h_{t-1}^{l}\right)\right).\end{aligned}$$

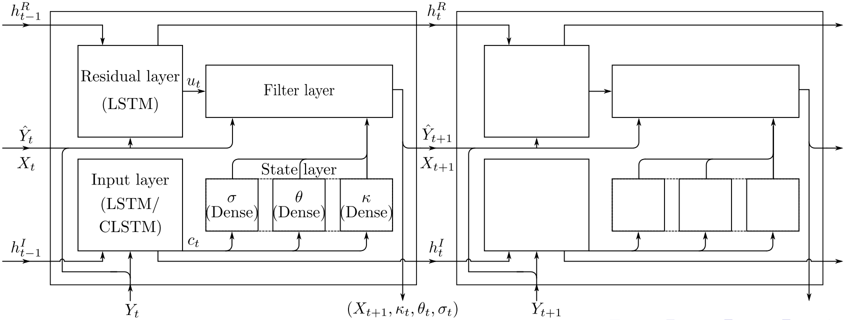
- Hidden value: $c_t^l \in \mathbb{R}^{1\times H}$.
- State layer: Three dense layers κ, θ and σ

$$\begin{aligned}\kappa(t, c_t^l) &: [0, T] \times \mathbb{R}^H \to \mathbb{R}^{d\times d}, & \kappa &= a_\kappa\left(\mathcal{W}_\kappa \cdot c_t^l + \mathcal{b}_\kappa\right), & a_\kappa(x) &= x,\\ \theta(t, c_t^l) &: [0, T] \times \mathbb{R}^H \to \mathbb{R}^{d}, & \theta &= a_\theta\left(\mathcal{W}_\theta \cdot c_t^l + \mathcal{b}_\theta\right), & a_\theta(x) &= \tanh(x),\\ \sigma(t, c_t^l) &: [0, T] \times \mathbb{R}^H \to \mathbb{R}^{d\times d}, & \sigma &= a_\sigma\left(\mathcal{W}_\sigma \cdot c_t^l + \mathcal{b}_\sigma\right), & a_\sigma(x) &= \tanh(x).\end{aligned}$$

- Residual cell:

$$
\begin{aligned}
e_t =& |Y_t - \hat{Y}_t|, \\
\bar{e}_t =& \mathrm{BN}(e_t), \\
(c_t^R, h_t^R) =& L_R\left(\bar{e}_t, \left(c_{t-1}^R, h_{t-1}^R\right)\right), \\
u_t =& D_R\left(c_t^R\right).
\end{aligned}
\tag{0.9}
$$

- RNN structure:



Loss function and other metrics

- Loss function: MSE of prediction error and arbitrage-free penalty

$$L(\vartheta) = \frac{1}{n}\sum_{i=1}^{n}\left|Y_i - \hat{Y}_i\right|^2 + \lambda\Lambda^{(p)}. \tag{0.10}$$

- Regularization (arbitrage-free penalty): p-norm of the excess return

$$\Lambda^{(p)} = \frac{1}{n}\sum_{i=1}^{n}\left\|\Lambda(t_i, T_j)\right\|_p = \frac{1}{n}\sum_{i=1}^{n}\sqrt[p]{\frac{1}{m}\sum_{j=1}^{m}\left|\Lambda(t_i, T_j)\right|^p}, \tag{0.11}$$

- Hit rate: Hitrate (spread) $= \frac{1}{NT}\sum_i^T\sum_j^N \mathbf{1}_{\left\{\left|Y(t_i,\tau_j) - \hat{Y}(t_i,\tau_j)\right| \leq \frac{1}{2}\text{spread}\right\}}$
- MAPE: Mean absolute prediction error, average of all samples

$$\frac{1}{N}\sum_{n=1}^{N}\left|\hat{Y}_n - Y_n\right|.$$

- RMAPE: Root mean absolute prediction error, average of all samples

$$\sqrt{\frac{1}{N}\sum_{n=1}^{N}\left|\hat{Y}_n - Y_n\right|^2}.$$

- Benchmark: use current yield curve to predict h-day-ahead yields and prices.

Implementation

- We consider 1-day-ahead and 5-day-ahead forecasting.
- The data used for h-day-ahead forecasting is batched from daily observations by every 5 days.
- By this method, we batch our data into h non-overlapping sequences and we can assume the h sequences as the observations on every Monday, Tuesday, Wednesday, Thursday and Friday.
- Restricted by the amount of data, we mixed the five distinct sequences as one training set instead of training them separately.
- To compare the forecasting results, we show the forecasting of yields in different terms of maturity: 3, 12, 36, 60, 120, 240, 360 months and group the forecasting of bond prices by maturities: $0 \sim 2$, $2 \sim 10$, $10 \sim 30$ year.

Implementation

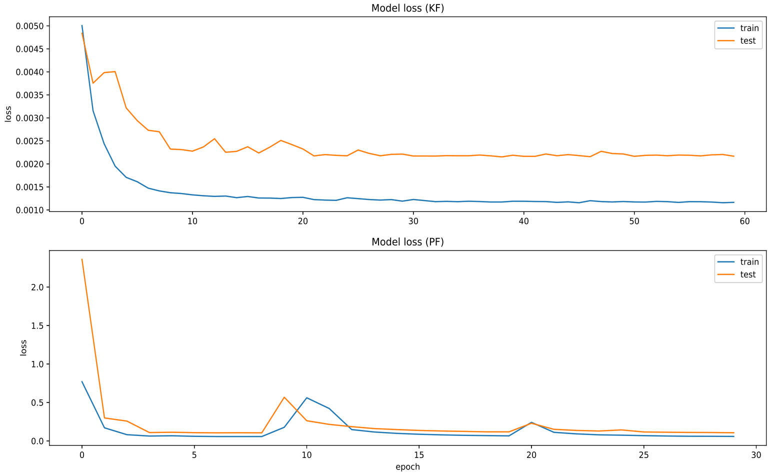
- We train three models: Kalman filter (KF), extended Kalman filter (EK) and particle filter (PF) with two arbitrage schemes $\lambda = 0$ and $\lambda = 1$ (arbitrage-regularized).
- The results from the Kalman filter contains only yield forecasts, hence we transform the predicted yields into prices of coupon bonds to examine the performance of our model.
- The extended Kalman filter provides both yields and prices forecasting.
- The particle filter predicts only prices and we apply an additional step to extract forecasted yields from the forecasted prices.

Implementation

- We run our models on Google Colab around 30$\sim$60 epochs which shows the optimal result without significant bias.
- The run times for the three filters are very different:
  - the Kalman filter runs with the fastest speed in a couple of minutes,
  - the extended Kalman filter takes a few seconds to finish 1 epoch depending on the amount of daily observations, and
  - the training time of the particle filter increases exponentially as the number of particles increase which can take a few hours with 300 particles.

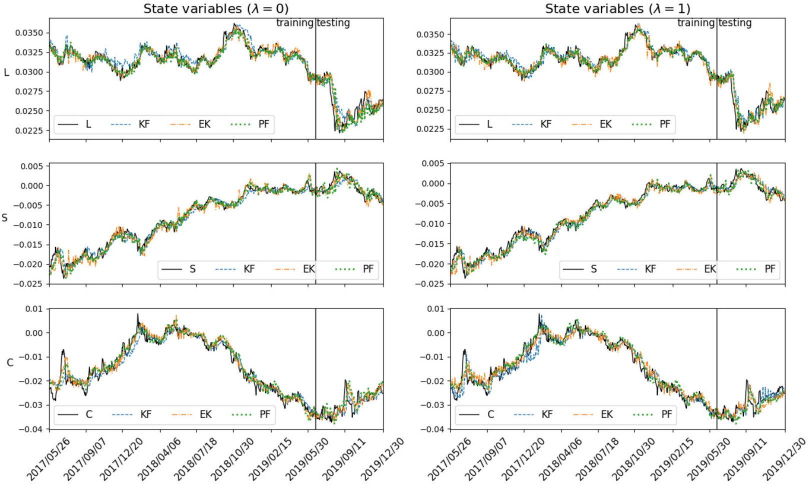
Implementation

Figure 3: Training result of U.S.Treasuries: Model loss



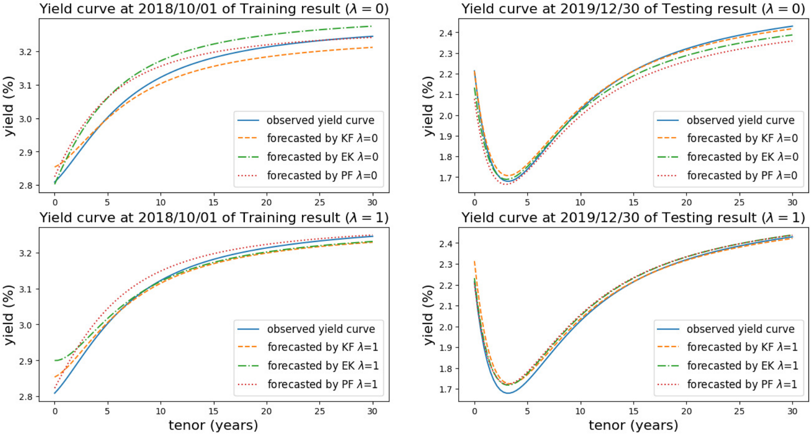
Forecasting of state variables

Figure 4: Result of U.S. Treasury: path of state variables of 5-day-ahead forecasting



Forecasting of yield curves

Figure 5: Result of U.S. Treasury: yield curves of 5-day-ahead forecasting



Prediction error of yield in h-day-ahead forecasting

Table 1: Testing result of U.S.Treasury: prediction error of yields (in bps)

Tenor	MAPE		RMSPE		STD		MAPE		RMSPE		STD	
	1-day	5-day	1-day	5-day	1-day	5-day	1-day	5-day	1-day	5-day	1-day	5-day
	KF(λ=0)						KF(λ=1)					
3M	2.9	4.55	3.83	6	3.78	5.76	2.86	9.21	3.88	11.2	3.81	6.68
1Y	2.85	4.7	3.78	6.25	3.74	6.23	2.78	5.7	3.61	7.51	3.57	6.66
3Y	3.66	7.06	4.9	9.29	4.89	9.2	3.51	7.14	4.52	9.35	4.51	9.06
5Y	3.95	7.99	5.21	10.3	5.21	10.1	3.73	7.91	4.83	10.1	4.83	9.85
10Y	3.92	8.01	4.99	10.2	4.99	9.93	3.72	7.8	4.74	9.99	4.73	9.83
20Y	3.81	7.6	4.89	9.8	4.88	9.48	3.82	9.23	4.78	12	4.77	10
30Y	3.83	7.51	4.97	9.75	4.95	9.43	3.89	10.3	4.91	13.3	4.89	10.3
	EK(λ=0)						EK(λ=1)					
3M	3.69	6.66	4.92	8.19	4.72	8.14	3.64	6.95	4.69	8.66	4.68	8.65
1Y	3.4	6.53	4.57	8.27	4.48	8.27	3.27	6.88	4.37	8.31	4.37	8.23
3Y	3.96	8.45	5.24	11	5.24	11	3.89	9.81	5.19	12.5	5.18	12.3
5Y	4.18	9.25	5.43	11.9	5.43	11.8	4.15	10.7	5.46	13.8	5.45	13.6
10Y	4.03	8.91	5.11	11.4	5.11	11.3	4.1	10.2	5.24	12.9	5.23	12.8
20Y	3.91	8.36	4.99	11	4.98	10.9	4.08	9.4	5.16	11.9	5.16	11.8
30Y	3.94	8.29	5.07	11	5.05	10.9	4.11	9.21	5.25	11.7	5.24	11.6
	PF(λ=0)						PF(λ=1)					
3M	4.83	8.33	6.24	10.1	6.21	9.81	5.89	10.4	8.1	13.4	8.1	10
1Y	4.01	7.66	5.07	9.89	5.05	9.78	4.38	8.56	5.9	11.3	5.89	9.44
3Y	3.97	9.23	5.15	12	5.15	12	3.67	9.19	4.84	11.8	4.83	11.6
5Y	4.15	9.57	5.34	12.5	5.34	12.5	3.87	9.66	5	12.4	4.98	12.3
10Y	4.01	9.16	5.05	11.8	5.05	11.8	3.8	9.23	4.8	11.9	4.79	11.9
20Y	3.93	9.07	5.01	11.8	5	11.8	3.77	8.69	4.79	11.8	4.79	11.8
30Y	4.01	9.3	5.13	12.2	5.12	12.1	3.85	8.64	4.92	12	4.92	12

Prediction error of price and hitrate

Table 2: Testing result of U.S.Treasury: price prediction error (in dollars)

Model	Yield MAPE		Yield RMSPE		Yield STD		Price MAPE		Price RMSPE		Price STD	
	1-day	5-day	1-day	5-day	1-day	5-day	1-day	5-day	1-day	5-day	1-day	5-day
KF(λ=0)	3.55	6.75	3.57	6.89	0.44	1.37	0.17	0.3	0.32	0.62	0.32	0.61
EK(λ=0)	3.86	8.11	3.87	8.17	0.27	1.01	0.18	0.35	0.35	0.74	0.35	0.73
PF(λ=0)	4.06	8.85	4.07	8.88	0.22	0.64	0.18	0.37	0.34	0.78	0.34	0.78
KF(λ=1)	3.42	7.5	3.45	7.62	0.41	1.35	0.17	0.34	0.31	0.79	0.31	0.79
EK(λ=1)	3.85	9.13	3.86	9.24	0.33	1.45	0.18	0.4	0.35	0.79	0.35	0.79
PF(λ=1)	4.03	9.11	4.06	9.13	0.54	0.52	0.17	0.37	0.33	0.79	0.33	0.78

Table 3: Testing result of U.S.Treasury: hit rate and percentage error

Spread	≤ 0.1		≤ 0.25		≤ 0.5		MPE		MPE		MPE	
Tenor(y)	$0 \sim 2$		$2 \sim 10$		$10 \sim 30$		$0 \sim 2$		$2 \sim 10$		$10 \sim 30$	
Model	1-day	5-day	1-day	5-day	1-day	5-day	1-day	5-day	1-day	5-day	1-day	5-day
Benchmark	79%	68%	80%	55%	42%	25%	0.065	0.085	0.157	0.32	0.683	1.439
KF(λ=0)	77%	69%	80%	58%	45%	19%	0.067	0.081	0.159	0.286	0.672	1.301
EK(λ=0)	78%	66%	76%	50%	38%	22%	0.066	0.087	0.168	0.343	0.725	1.533
PF(λ=0)	79%	65%	77%	49%	37%	15%	0.065	0.091	0.167	0.362	0.722	1.614
KF(λ=1)	77%	67%	81%	57%	42%	22%	0.066	0.086	0.152	0.284	0.681	1.706
EK(λ=1)	80%	65%	77%	44%	38%	19%	0.065	0.09	0.168	0.399	0.741	1.673
PF(λ=1)	77%	64%	79%	49%	42%	25%	0.067	0.096	0.157	0.365	0.693	1.58

Average excess return and Effective sample size

Figure 6: Average excess return of 1-day-ahead forecasting

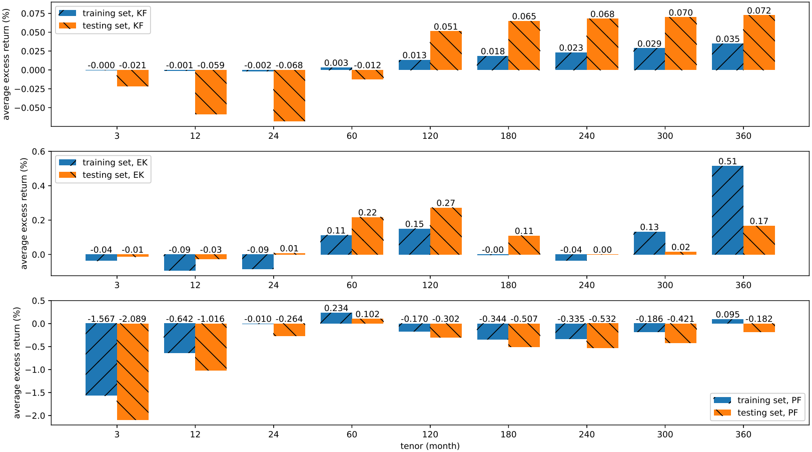
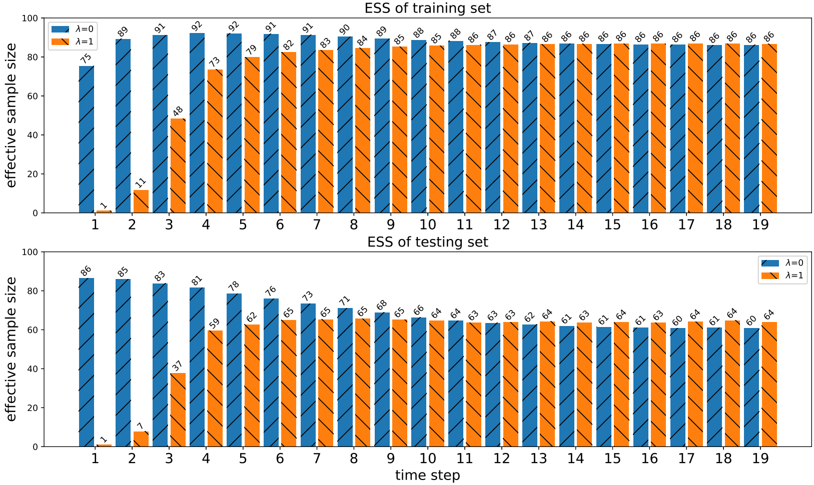


Figure 7: Effective sample size of 1-day-ahead forecasting



Price error distribution of 1-day-ahead forecasting

Figure 8: Price error distribution

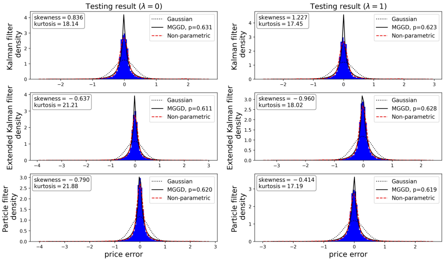
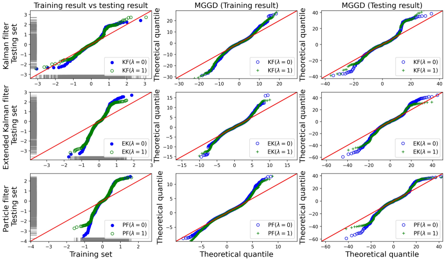


Figure 9: QQ-plot (price error)



Decrease the value of p in MGGD: small p gives fat tail and steep center.

Yield error distribution of 1-day-ahead forecasting

Figure 10: Result of U.S. Treasury: Yield error distribution

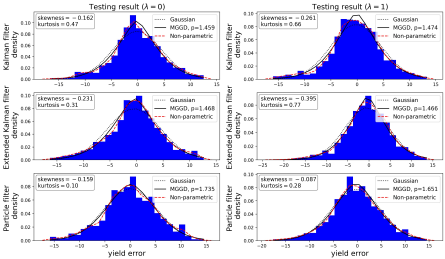
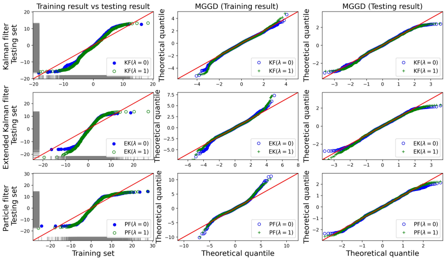


Figure 11: Result of U.S. Treasury: QQ-plot (yield error)



State Parameters: without AF regularization

Figure 12: U.S.Treasuries: State parameters (Kalman filter)

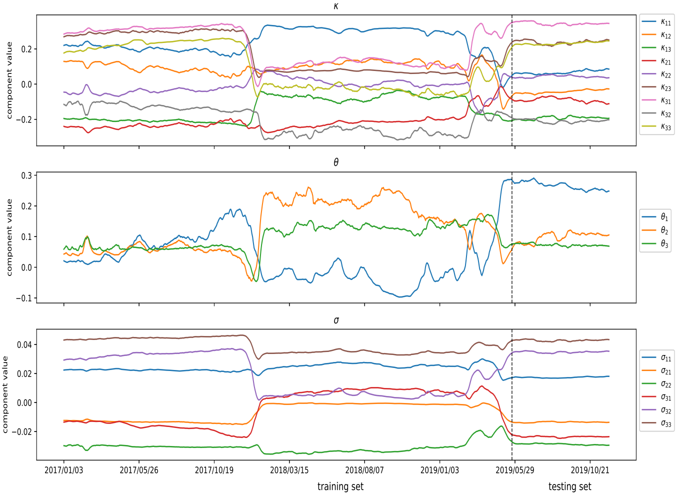
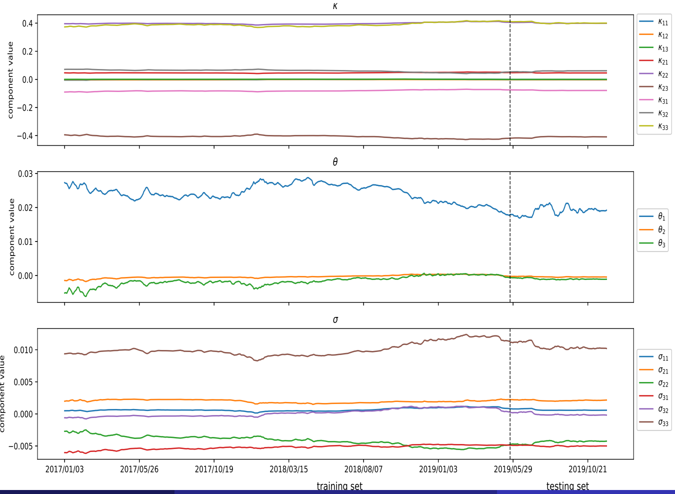


Figure 13: U.S.Treasuries: State parameters (Kalman filter + arbitrage regularization)



Corporate Bonds

- We apply our models to 10 corporate issuers.
- The following table shows the 5-day-ahead forecasting results on data from ten corporate bond issuers with the predicted corporate spread calculated by subtracting the the predicted Treasury yield from the predicted corporate yield and comparing to the observed value.
- For the forecasting results of corporate data, we show the predicted spread errors (predicted corporate yields - predicted Treasury yields) are less 14 bps in 5-day-ahead forecasting.
- Since credit risk factors are not included and the corporate data contains only around 10 to 30 daily bonds which is too few, the forecasting performance of corporate data is not comparable to that of Treasury data.
- Since yields data contains more information than the corporate bonds, we find that the model with Kalman filter significant outperforms the model with extended Kalman filter or particle filter.

Corporate Bonds

Table 4: Testing result of 5-day-ahead forecasting: spread error (bps) and price error (dollar)

Ticker	MAPE		STDV		MPE	Hitrate	Ticker	MAPE		STDV		MPE	Hitrate
Model	Spread	Price	Spread	Price	%	≤ 0.25	Model	Spread	Price	Spread	Price	%	≤ 0.25
AAPL							AAPL						
KF (λ=0)	5.27	0.241	2.54	0.353	0.231	68.1%	KF (λ=1)	6.7	0.246	3.02	0.359	0.348	59.7%
EK (λ=0)	11.47	0.397	5.08	0.635	0.379	52.9%	EK (λ=1)	10.52	0.355	3.51	0.523	0.341	53.0%
PF (λ=0)	10.4	0.259	3.44	0.386	0.248	65.8%	PF (λ=1)	9.44	0.351	2.58	0.533	0.288	60.6%
C							C						
KF (λ=0)	9.93	0.447	2.92	0.713	0.404	45.5%	KF (λ=1)	9.34	0.449	1.92	0.704	0.390	46.0%
EK (λ=0)	10.66	0.454	3.27	0.726	0.411	43.8%	EK (λ=1)	13	0.491	5.06	0.793	0.408	45.3%
PF (λ=0)	11.05	0.49	2.44	0.816	0.440	45.1%	PF (λ=1)	14.03	0.534	6.45	0.898	0.412	44.1%
DIS							DIS						
KF (λ=0)	7.83	0.365	3.46	0.847	0.341	62.0%	KF (λ=1)	11.76	0.369	3.48	0.587	0.334	62.5%
EK (λ=0)	11.24	0.372	4.81	0.67	0.351	53.2%	EK (λ=1)	11.02	0.394	3.94	0.757	0.352	55.2%
PF (λ=0)	11.33	0.383	4.44	0.721	0.358	54.7%	PF (λ=1)	9.61	0.382	3.52	0.812	0.363	54.0%
GS							GS						
KF (λ=0)	8.75	0.426	2.89	0.615	0.388	47.2%	KF (λ=1)	8.66	0.428	1.71	0.604	0.402	45.6%
EK (λ=0)	9.16	0.437	1.96	0.649	0.402	47.3%	EK (λ=1)	10.81	0.434	2.72	0.658	0.402	48.0%
PF (λ=0)	10.24	0.433	3.06	0.653	0.397	47.7%	PF (λ=1)	11.03	0.475	2.94	0.727	0.400	47.1%
JNJ							JNJ						
KF (λ=0)	7.38	0.454	3.78	0.706	0.412	47.0%	KF (λ=1)	8.06	0.429	3.46	0.645	0.400	47.4%
EK (λ=0)	10.61	0.54	6.13	0.879	0.496	40.4%	EK (λ=1)	10.21	0.541	3.58	0.882	0.518	42.0%
PF (λ=0)	9.96	0.578	4.01	0.937	0.527	41.6%	PF (λ=1)	11.01	0.607	3.9	1.012	0.475	41.9%

Corporate Bonds

Table 5: Testing result of 5-day-ahead forecasting: spread error (bps) and price error (dollar)

Ticker	MAPE		STDV		MPE	Hitrate	Ticker	MAPE		STDV		MPE	Hitrate
Model	Spread	Price	Spread	Price	%	≤ 0.25	Model	Spread	Price	Spread	Price	%	≤ 0.25
JPM							JPM						
KF (λ=0)	6.46	0.346	1.77	0.616	0.307	58.7%	KF (λ=1)	8.49	0.45	4.12	0.914	0.324	55.9%
EK (λ=0)	10.07	0.473	3.95	0.909	0.412	52.1%	EK (λ=1)	11.31	0.508	3.38	0.96	0.398	48.3%
PF (λ=0)	10.62	0.491	3.58	0.914	0.429	50.9%	PF (λ=1)	12.63	0.482	4.97	0.926	0.430	50.0%
MSFT							MSFT						
KF (λ=0)	5.62	0.343	2.77	0.492	0.325	52.5%	KF (λ=1)	8.34	0.448	4.96	0.807	0.331	52.3%
EK (λ=0)	10.56	0.441	3.37	0.653	0.419	45.9%	EK (λ=1)	9.9	0.429	2.84	0.662	0.434	43.4%
PF (λ=0)	11.6	0.393	4.14	0.594	0.373	50.1%	PF (λ=1)	10.84	0.433	3.6	0.67	0.406	47.4%
T							T						
KF (λ=0)	9.41	0.45	5.62	1.022	0.398	57.1%	KF (λ=1)	10.26	0.407	5.11	0.85	0.370	59.6%
EK (λ=0)	10.56	0.489	4.59	0.951	0.440	49.8%	EK (λ=1)	14.14	0.611	5	1.057	0.431	48.7%
PF (λ=0)	12.81	0.389	5.31	0.75	0.353	57.5%	PF (λ=1)	14.04	0.53	6.55	1.032	0.361	57.5%
UNH							UNH						
KF (λ=0)	8.34	0.404	3.63	0.881	0.377	54.2%	KF (λ=1)	10.15	0.433	5.18	0.909	0.343	55.5%
EK (λ=0)	9.23	0.364	4.18	0.588	0.344	53.4%	EK (λ=1)	11.68	0.376	4.65	0.615	0.361	53.1%
PF (λ=0)	10.96	0.378	3.09	0.744	0.353	56.5%	PF (λ=1)	10.34	0.371	3.89	0.602	0.349	57.0%
WFC							WFC						
KF (λ=0)	7.03	0.368	2.82	0.778	0.330	68.0%	KF (λ=1)	10.64	0.442	3.72	0.878	0.337	66.5%
EK (λ=0)	12.14	0.56	3.38	1.117	0.501	55.8%	EK (λ=1)	13.25	0.562	2.8	1.166	0.592	53.8%
PF (λ=0)	12.77	0.538	3.34	1.029	0.487	54.5%	PF (λ=1)	13.71	0.583	3.52	1.12	0.560	50.2%

Conclusion

- Arbitrage-free restriction improves the performance in short time forecasting.
- Arbitrage-free restriction downgrades the performance in long time forecasting.
- Easy to quantify and restrict the excess return.
- Yield error and bond price error are not subject to Gaussian pattern. They are more likely to be generalized multi-Gaussian or non-parametric distribution.
- Kalman filter is very efficient (between training and testing) in yield prediction.
- Particle filter with importance sampling is very efficient (non-degeneracy) in bond price prediction.
- Our model with regularization is robust for both training and testing data.

References

[1] A. Ang and M. Piazzesi. A no-arbitrage vector autoregression of term structure dynamics with macroeconomic and latent variables. *Journal of Monetary Economics*, 50(4):745–787, 2003.

[2] T. Björk and L. Svensson. On the existence of finite-dimensional realizations for nonlinear forward rate models. *Mathematical Finance*, 11(2):205–243, 2001.

[3] J. H. Christensen, F. X. Diebold, and G. D. Rudebusch. The affine arbitrage-free class of Nelson–Siegel term structure models. *Journal of Econometrics*, 164(1):4–20, 2011.

[4] F. X. Diebold and C. Li. Forecasting the term structure of government bond yields. *Journal of Econometrics*, 130(2):337–364, 2006.

[5] A. Kratsios and C. Hyndman. Deep arbitrage-free learning in a generalized HJM framework via arbitrage-regularization. *Risks*, 8(40):1–30, 2017.

[6] F. Pascal, L. Bombrun, J.-Y. Tourneret, and Y. Berthoumieu. Parameter estimation for multivariate generalized gaussian distributions. *IEEE Transactions on Signal Processing*, 61(23):5960–5971, 2013.