Article

Not peer-reviewed version

Particle Filtering Estimation of Regime Switching Factor Model and Its Application in Statistical Arbitrage Strategy

Yu Mu * and Robert J. Frey *

Posted Date: 24 July 2025

doi: 10.20944/preprints202507.2022.v1

Keywords: regime switching; statistical factor model; particle filtering; statistical arbitrage

 doi:10.20944/preprints202507.2022.v1

Article

Particle Filtering Estimation of Regime Switching Factor Model and Its Application in Statistical Arbitrage Strategy

Yu Mu * and Robert Frey *

Stony Brook University
* Correspondence: ymu0117@gmail.com (Y.M.); robert.frey@stonybrook.edu (R.F.)

Abstract

Statistical factor models are widely applied across various domains of the financial industry, including risk management, portfolio selection, and statistical arbitrage strategies. However, conventional factor models often rely on unrealistic assumptions and fail to account for the fact that financial markets operate under multiple regimes. In this paper, we propose a regime-switching factor model estimated using a particle filtering algorithm which is a Monte Carlo-based method well-suited for handling nonlinear and non-Gaussian systems. Our empirical results show that incorporating regime-switching dynamics significantly enhances the model's ability to detect structure breaks and adapt to evolving market conditions. This leads to improved performance and reduced drawdown in the equity statistical arbitrage strategies.

Keywords: regime switching; statistical factor model; particle filtering; statistical arbitrage

1. Introduction

Factor analysis plays a critical role in various areas of quantitative finance, including asset pricing, risk management, portfolio selection, and statistical arbitrage strategies, etc.. Broadly speaking, factor models can be categorized into two types: explicit factor models, which fall under supervised learning, and statistical factor models, which are typically unsupervised. Compared to explicit factor models such as Barra risk model [1], statistical factor models have two key advantages.

First, by construction, statistical factor models are more numerically stable and can capture the underlying covariance structure using fewer factors. Second, they are asset-class agnostic and therefore more broadly applicable across different markets. Given these strengths, this work focuses on the estimation and application of statistical factor models.

Statistical factor models has been developed as early as the 1930s. Scholars started to use factor models based on principal component analysis (PCA) back then [2]. PCA based factor models failed to provide a parametric model for the observations and perform poorly when within group variation dominates between group variation [3,4]. In the 1980s, maximum likelihood (ML) factor analysis estimated via the EM algorithm was introduced by Rubin and Thayer [5]. This was further generalized in the 1990s by Ghahramani and Hinton [6], who proposed the Mixture of Factor Analyzers. Their approach explains the hidden factors' structure by combining clustering and local dimension reduction techniques. Mixture of factor analyzers works better than classical ML factor analysis due to its ability of segmenting data into different market regimes, each with its own factor structure. It handles the market regime changes in the covariance structure and hence has a more realistic assumption than the "one-size-fits-all" ML factor models.

Despite this advancement, within each regime, it still uses classical ML factor model which assumes i.i.d. hidden factors and Gaussian innovations. Dynamic factor model has a recursive structure for the hidden factor returns and can better model the factor return process [7–10]. This paper

 doi:10.20944/preprints202507.2022.v1

focus on the estimation of dynamic factor with regime switching factor loading that is governed by first order Markov Chain. The estimation relies on Particle filtering techniques [11,12], a class of Monte Carlo methods designed for solving the recursive Bayesian inference when analytical solutions such as the Kalman filter or Baum-Welch algorithm are infeasible.

Although particle filtering techniques is well developed when model parameter values are known, parameter learning in more complex settings remains challenging. Specifically, when parameters evolve according to a stochastic process rather than being constant, such as the applications in finance, an adaptive parameter learning algorithm becomes critical. To learn the parameter values, one early approach by Liu [13] incorporates parameters into the hidden state vector, but this often exacerbates the degeneracy issue in particle filtering. More computationally robust approaches have been introduced in [14,15]. Instead of combining parameter estimation to the learning algorithm they marginalize over parameters to reduce the estimation variance of their system. In this work, we implemented a particle filtering algorithm inspired by [14,15], and evaluate the estimation of regime switching dynamic factor model in the context of an equity statistical arbitrage strategy.

The rest of paper is organized as follows: Section 2 introduces the state space model and its estimation methods, including Kalman filter, EM algorithm, Particle filter, and the more recent development in the parameter learning - particle learning algorithm. Section 3 presents statistical factor models such as the MLE factor model, mixture of factor analyzers and the regime-switching dynamic factor model. We highlight the evolution of statistical factor models over recent decades and provides a simulation study demonstrating the effectiveness of particle learning algorithm for estimating the regime-switching dynamic factor model. Section 4 presents emprical studies applying the regime switching dynamic factor model in the context of equity stat-arb strategy and compares its performance to the conventional MLE factor model. Finally, Section **??** concludes the paper.

2. State Space Model and Its Estimation

A state space model is a framework used to describe the evolution of a dynamical system over time, particularly when the system is only partially observed. It has numerous applications in finance, such as time series prediction (alpha generation), statistical factor analysis (risk modeling), portfolio selection, etc.. In the most general form, a state space model can be represented by:

$$\mathbf{x}_t = f_t(\mathbf{x}_{t-1}, \mathbf{w}_t) \tag{1}$$

$$\mathbf{y}_t = g_t(\mathbf{x}_t, \mathbf{v}_t) \tag{2}$$

where $\mathbf{x}_t$ denote the hidden states with dimension k, and $\mathbf{y}_t$ the observations with dimension s. The terms $\mathbf{w}_t$ and $\mathbf{v}_t$ represent the noise terms for the evolution equation (1) and the observation equation (2), with dimensions w and v respectively. The functions f_t and g_t are general nonlinear functions defining the state transition and observation processes. When the parameter values of the above equations are known, inference about the hidden states can be efficiently performed using the recursive Bayesian algorithm [12], which is summarized as Algorithm 1:

Algorithm 1 Recursive Bayesian Algorithm

1. Prediction:

$$p(\mathbf{x}_t \mid \mathbf{y}_{1:t-1}) = \int p(\mathbf{x}_t \mid \mathbf{x}_{t-1}) p(\mathbf{x}_{t-1} \mid \mathbf{y}_{1:t-1}) d\mathbf{x}_{t-1} \tag{3}$$

2. Update:

$$p(\mathbf{x}_t \mid \mathbf{y}_{1:t}) = \frac{p(\mathbf{y}_t \mid \mathbf{x}_t) p(\mathbf{x}_t \mid \mathbf{y}_{1:t-1})}{p(\mathbf{y}_t \mid \mathbf{y}_{1:t-1})} \tag{4}$$

 doi:10.20944/preprints202507.2022.v1

where the denominator in equation (4) is another intergral needs to be solved in this problem:

$$p(\mathbf{y}_t \mid \mathbf{y}_{1:t-1}) = \int p(\mathbf{y}_t \mid \mathbf{x}_t) p(\mathbf{x}_t \mid \mathbf{y}_{1:t-1}) d\mathbf{x}_t \tag{5}$$

2.1. Kalman Filter and Smoother

When the dynamical system described by equations (1) and (2) is linear, time-invariant, and finite-dimensional, functions f_t, and g_t can be expressed in the following form:

$$\mathbf{x}_t = \mathbf{B}\mathbf{x}_{t-1} + \mathbf{w}_t \tag{6}$$
$$\mathbf{y}_t = \mathbf{A}\mathbf{x}_t + \mathbf{v}_t \tag{7}$$

and if we further assume $\mathbf{w}_t \sim N(0, \mathbf{Q})$, $\mathbf{v}_t \sim N(0, \mathbf{R})$, then Kalman filter can be derived to solve equations (6) and (7) analytically as described in [12] and Algorithm 2.

Algorithm 2 Kalman Filter

1. Prediction:

$$\mathbf{m}_{t|t-1} = \mathbf{B}\mathbf{m}_{t-1|t-1} \tag{8}$$
$$\mathbf{C}_{t|t-1} = \mathbf{Q} + \mathbf{B}\mathbf{C}_{t-1|t-1}\mathbf{B}^T \tag{9}$$

2. Update:

$$\mathbf{m}_{t|t} = \mathbf{m}_{t|t-1} + \mathbf{K}_t\left(\mathbf{y}_t - \mathbf{A}\mathbf{m}_{t|t-1}\right) \tag{10}$$
$$\mathbf{C}_{t|t} = \mathbf{C}_{t|t-1} - \mathbf{K}_t\mathbf{A}\mathbf{C}_{t|t-1} \tag{11}$$

where $\mathbf{K}_t$ is the Kalman gain and can be computed by:

$$\mathbf{S}_t = \mathbf{A}\mathbf{C}_{t|t-1}\mathbf{A}^T + \mathbf{R} \tag{12}$$
$$\mathbf{K}_t = \mathbf{C}_{t|t-1}\mathbf{A}^T\mathbf{S}_t^{-1} \tag{13}$$

When estimating hidden states in offline mode (i.e., when the full time series data is available) or when performing EM algorithm to estimate model parameters, the Kalman smoother is typically used to refine those state estimates using future observations as well (See Algorithm 3).

Algorithm 3 Kalman Smoother

$$\mathbf{m}_{t|T} = \mathbf{m}_{t|t} + \mathbf{K}_{t|T}\left(\mathbf{m}_{t+1|T} - \mathbf{B}\mathbf{m}_{t|t}\right) \tag{14}$$
$$\mathbf{C}_{t|T} = \mathbf{C}_{t|t} + \mathbf{K}_{t|T}\left(\mathbf{C}_{t+1|T} - \mathbf{C}_{t+1|t}\right)\mathbf{K}_{t|T}^T \tag{15}$$

where $\mathbf{K}_{t|T} = \mathbf{C}_{t|t}\mathbf{B}^T\mathbf{C}_{t+1|t}^{-1}$ is the Kalman smoothing gain.

2.2. Parameter Estimation using EM algorithm

The Kalman filter algorithm introduced above assumes parameters of the system are known and fixed. When parameters are unknown, the EM algorithm [6] can be used to estimate these unknown parameters in a state-space form:

 doi:10.20944/preprints202507.2022.v1

Algorithm 4 EM algorithm

1. E-Step:

$$E[\mathbf{x}_t \mid \mathbf{y}_{1:T}] = \mathbf{m}_{t|T} \tag{16}$$

$$E\left[\mathbf{x}_t\mathbf{x}_t^T \mid \mathbf{y}_{1:T}\right] = \mathbf{C}_{t|T} + \mathbf{m}_{t|T}\mathbf{m}_{t|T}^T \tag{17}$$

$$E\left[\mathbf{x}_t\mathbf{x}_{t-1}^T \mid \mathbf{y}_{1:T}\right] = \mathbf{C}_{t,t-1|T} + \mathbf{m}_{t|T}\mathbf{m}_{t-1|T}^T \tag{18}$$

2. M-Step:

$$\hat{\mathbf{A}} = \left(\sum_{t=1}^{T} \mathbf{y}_t E[\mathbf{x}_t \mid \mathbf{y}_{1:T}]^T\right)\left(\sum_{t=1}^{T} E\left[\mathbf{x}_t\mathbf{x}_t^T \mid \mathbf{y}_{1:T}\right]\right)^{-1} \tag{19}$$

$$\hat{\mathbf{R}} = \frac{1}{T}\sum_{t=1}^{T}\left(\mathbf{y}_t\mathbf{y}_t^T - \hat{\mathbf{A}}E[\mathbf{x}_t \mid \mathbf{y}_{1:T}]\mathbf{y}_t^T\right) \tag{20}$$

$$\hat{\mathbf{B}} = \left(\sum_{t=2}^{T} E\left[\mathbf{x}_t\mathbf{x}_{t-1}^T \mid \mathbf{y}_{1:T}\right]\right)\left(\sum_{t=2}^{T} E[\mathbf{x}_{t-1} \mid \mathbf{y}_{1:T}]\right)^{-1} \tag{21}$$

$$\hat{\mathbf{Q}} = \frac{1}{T-1}\left(\sum_{t=2}^{T} E\left(\mathbf{x}_t\mathbf{x}_t^T \mid \mathbf{y}_{1:T}\right) - \hat{\mathbf{A}}\sum_{t=2}^{T} E\left[\mathbf{x}_t\mathbf{x}_{t-1}^T \mid \mathbf{y}_{1:T}\right]\right) \tag{22}$$

$$\mathbf{m}_1 = E[\mathbf{x}_1 \mid \mathbf{y}_{1:T}] \tag{23}$$

$$\mathbf{C}_1 = E\left[\mathbf{x}_1\mathbf{x}_1^T \mid \mathbf{y}_{1:T}\right] - E[\mathbf{x}_1 \mid \mathbf{y}_{1:T}]E[\mathbf{x}_1 \mid \mathbf{y}_{1:T}]^T \tag{24}$$

where $\mathbf{m}_{t|T}$, $\mathbf{C}_{t|T}$ will be estimated by Kalman filter and smoother. $\mathbf{C}_{t,t-1|T}$ can be updated by backward recursions:

$$\mathbf{C}_{t,t-1|T} = \mathbf{C}_{t|t}\mathbf{K}_{t-1|T}^T + \mathbf{K}_{t|T}\left(\mathbf{C}_{t+1,t|T} - \mathbf{B}\mathbf{C}_{t|t}\right)\mathbf{K}_{t-1|T}^T \tag{25}$$

which is initialized by $\mathbf{C}_{T,T-1|T} = (\mathbf{I} - \mathbf{K}_T\mathbf{A})\mathbf{B}\mathbf{C}_{T-1|T-1}$.

2.3. Particle Filter

When the dynamical system (1) and (2) do not admit a linear-Gaussian representation as in (6), and (7), deriving an analytical solution for the recursive Bayesian inference becomes challenging. In such cases, particle filtering techniques has been a very popular approach for solving nonlinear dynamical system, including models like the stochastic volatility model [14]. In this section, we introduce several baseline particle filtering algorithms.

2.3.1. SIS Particle Filter

Sequential Importance Sampling (SIS) particle filter has been introduced as early as 1990s [16]. The main idea is to approximate the joint posterior density at time t using importance sampling method:

$$p(\mathbf{x}_{1:t} \mid \mathbf{y}_{1:t}) \approx \sum_{i=1}^{N} W_t^i \delta\left(\mathbf{x}_{1:t} - \mathbf{x}_{1:t}^i\right) \tag{26}$$

where $\delta(\cdot)$ is the Dirac delta function, and W_t^i is the importance weight for particle i, given by:

$$W_t^i \propto \frac{p(\mathbf{x}_{1:t} \mid \mathbf{y}_{1:t})}{q(\mathbf{x}_{1:t} \mid \mathbf{y}_{1:t})} \tag{27}$$

 doi:10.20944/preprints202507.2022.v1

where $p(\mathbf{x}_{1:t} \mid \mathbf{y}_{1:t})$ is the target distribution and $q(\mathbf{x}_{1:t} \mid \mathbf{y}_{1:t})$ is the importance (or proposal) distribution. A sequential estimation algorithm can be derived if we assume the joint importance density has the following factorization:

$$q(\mathbf{x}_{1:t} \mid \mathbf{y}_{1:t}) = q(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{y}_t) q(\mathbf{x}_{1:t-1} \mid \mathbf{y}_{1:t-1}) \tag{28}$$

This factorization allows sequential sampling of the state $\mathbf{x}_t$ given the previous state and current observation, which forms the foundation of the SIS particle filter as in Algorithm 5:

Algorithm 5 SIS Particle Filter

1. Draw $\mathbf{x}_t^i \sim q(\mathbf{x}_t \mid \mathbf{x}_{t-1}^i, \mathbf{y}_t)$
2. Assign the particle $\mathbf{x}_t^i$ a weight, $W_t^i \propto W_{t-1}^i \frac{p(\mathbf{y}_t|\mathbf{x}_t^i) p(\mathbf{x}_t^i|\mathbf{x}_{t-1}^i)}{q(\mathbf{x}_t^i|\mathbf{x}_{t-1}^i, \mathbf{y}_t)}$

2.3.2. SIR Particle Filter

The SIS particle filter often fails in practice due to the weight degeneracy issue. The variance of the weight W_t^i in SIS filter will increase over time until only one particle left with large weight. To alleviate this issue, we can resample (with replacement) the particles based on their weights. By doing so we can replicate those particles with large weights, eliminate particles with small weights, and hence reduce the variance of weights W_t^i [11].

Moreover, It has been shown the optimal importance density for a generic particle filter algorithm is $p(\mathbf{x}_t \mid \mathbf{x}_{t-1}^i, \mathbf{y}_t)$ [17] and if we are able to choose the importance density close to the optimal importance density we can further alleviate the weight degeneracy issue. SIR particle filter uses $p(\mathbf{x}_t \mid \mathbf{x}_{t-1}^i)$ as important density and applying resampling at each step as shown in the Algorithm 6:

Algorithm 6 SIR Particle Filter

1. Draw $\mathbf{x}_t^i \sim p(\mathbf{x}_t \mid \mathbf{x}_{t-1}^i)$
2. Assign the particle $\mathbf{x}_t^i$ a weight, $W_t^i \propto p(\mathbf{y}_t \mid \mathbf{x}_t^i)$
3. Resampling based on weight W_t^i

Although SIR filter improves the weight degeneracy problem present in the SIS filter, it still has its own weakness [18]. First, if there is an outlier at time t, the weights W_t^i will unevenly distributed and the filter will become imprecise. Second, the predictive density $p(\mathbf{x}_t \mid \mathbf{x}_{t-1}^i)$ often fails to capture tail behavior accurately, causing some particles to drift into low-likelihood regions during the prediction step (2).

2.3.3. Auxiliary Particle Filter

To address the limitations of the SIR filter, the auxiliary particle filter proposed by [18] introduces a lookahead step - also known as the auxiliary variable mechanism. This approach uses a predictive weighting scheme (See Algorithm 7 step 1) that estimates the likelihood of the upcoming observation given the predicted state, a process often referred to as lookahead weighting.

Algorithm 7 Auxiliary Particle Filter

1. (Resample) Draw $k^i \sim Multi\{W_t^1, \ldots, W_t^N\}$,where $W_t^i \propto p(\mathbf{y}_t \mid \mu(\mathbf{x}_{t-1}^i))$.
2. (Propagate) Draw $\tilde{\mathbf{x}}_t^i \sim p\left(\mathbf{x}_t \mid \mathbf{x}_{t-1}^{k^i}\right)$.
3. (Resample) Draw $\mathbf{x}_t^i \sim Multi\{W_t^1, \ldots, W_t^N\}$, where $W_t^t \propto p(\mathbf{y}_t \mid \tilde{\mathbf{x}}_t^i) / p(\mathbf{y}_t \mid \mu(\mathbf{x}_{t-1}^i))$.

 doi:10.20944/preprints202507.2022.v1

where $\mu(\mathbf{x}_{t-1}^i)$ denotes the mean, median, or mode of the distribution of $\mathbf{x}_t$ given $\mathbf{x}_{t-1}^i$. When the observation equation is informative in dynamical system, Auxiliary particle filter usually will reduce variance in particle weights and hence perform better.

2.4. Particle Filter with Parameter Learning

The particle filter algorithms discussed so far assume that model parameters are known. However, in many applications, model parameters are typically unknown. Learning the static parameter values - or more broadly - identifying the coefficient evolution process - is a critical aspect of system identification. The integration of parameter learning into the particle filtering framework has been extensively studied [19].

One idea is to treat parameters as additional hidden state variables in the particle filtering framework, but this approach often leads to severe degeneracy issue. To address this issue, [13] proposed introducing an artificial evolutional model for the parameters and augment the hidden state space by including them.

Another idea is to integrate out (or marginalize out) the parameters whenever feasible [14,20,21]. The benefits of the marginalized particle filter has been discussed in [22].

Building on these ideas, Carvalho et al. [15] introduced particle learning approach, a noval particle filter with the embedded parameter learning step. Particle learning method marginalize both the parameters and the hidden states by tracking only the sufficient statistics of them, which updated later using recursive Bayesian algorithm and Kalman filter, respectively. It is also constructed under the Auxiliary particle filter framework, and the performance improvement has been demonstrated in [15,23].

In our paper, we develop a simplified version of the particle learning algorithm to estimate regime switching dynamic factor model.

2.4.1. Particle Learning

Particle learning algorithm was introduced by Carvalho et al. in [15], extending the idea of auxiliary particle filter to incorporate parameter learning. While the overall filtering scheme is the same as auxiliary particle filter. the key innovation lies in tracking the "essential state" $\mathbf{z}_t$.

Specifically, $\mathbf{z}_t$ includes:

- the sufficient statistics of the parameter vector $\mathbf{s}_t$,
- the sufficient statistics of hidden states $\mathbf{x}_t$, $\mathbf{s}_t^x$, and
- the current value of the parameter vector, θ_t:

$$\mathbf{z}_t = \{\mathbf{s}_t, \mathbf{s}_t^x, \theta_t\}.$$

The reason we include parameter value in essential state is because for evaluating likelihood and drawing samples from predictive density we need to use parameter values. However, there values are not directly sampled at each step, which most likely will cause degeneracy issue; instead, they are inferred offline from their associated sufficient statistics. This approach is also referred to as marginalization by simulation.

The derivation of this filter start with the recursive relation:

$$p(\mathbf{z}_t \mid \mathbf{y}_{1:t}) \propto \int p(\mathbf{y}_t \mid \mathbf{z}_{t-1}) p(\mathbf{z}_t \mid \mathbf{z}_{t-1}, \mathbf{y}_t) p(\mathbf{z}_{t-1} \mid \mathbf{y}_{1:t-1}) d\mathbf{z}_{t-1}$$

where

$$\begin{aligned} p(\mathbf{z}_t \mid \mathbf{z}_{t-1}, \mathbf{y}_t) = \int & p(\mathbf{s}_t \mid \mathbf{s}_{t-1}, \mathbf{x}_t, \mathbf{y}_t) p\left(\mathbf{s}_t^x \mid \mathbf{s}_{t-1}^x, \mathbf{x}_t, \mathbf{y}_t\right) \\ & \cdot p\left(\mathbf{x}_t \mid \mathbf{s}_{t-1}^x, \mathbf{y}_t\right) d\mathbf{x}_t \end{aligned}$$

and

$$p\left(\mathbf{x}_t \mid \mathbf{s}_{t-1}^x, \mathbf{y}_t\right) = \int p(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{y}_t) p\left(\mathbf{x}_{t-1} \mid \mathbf{s}_{t-1}^x, \mathbf{y}_t\right) d\mathbf{x}_{t-1}$$

 doi:10.20944/preprints202507.2022.v1

So given the particles of $\mathbf{s}_{t-1}^x$ we can construct Monte Carlo estimate for $p(\mathbf{x}_t \mid \mathbf{s}_{t-1}^x, \mathbf{y}_t)$ by sampling from $p\left(\mathbf{x}_{t-1} \mid (\mathbf{s}_{t-1}^x)^i, \mathbf{y}_t\right)$ and $p(\mathbf{x}_t \mid \mathbf{x}_{t-1}^i, \mathbf{y}_t)$ respectively. After that we can update $\mathbf{s}_t$ and $\mathbf{s}_t^x$ deterministically. Particle learning method can be summarized as follows:

Algorithm 8 Particle Learning

1. (Resample) Draw $k^i \sim Multi\{W_t^1, \ldots, W_t^N\}$, where $W_t^i \propto p(\mathbf{y}_t \mid \mathbf{z}_{t-1}^i)$
2. (Propagate) Draw $\mathbf{x}_{t-1}^i \sim p\left(\mathbf{x}_{t-1} \mid (\mathbf{s}_{t-1}^x)^{k^i}, \theta_{t-1}^{k^i}, \mathbf{y}_t\right)$
3. (Propagate) Draw $\mathbf{x}_t^i \sim p\left(\mathbf{x}_t \mid \mathbf{x}_{t-1}^i, \theta_{t-1}^{k^i}, \mathbf{y}_t\right)$
4. (Update) $\mathbf{s}_t^i = \mathcal{S}\left(\mathbf{s}_{t-1}^{k^i}, \mathbf{x}_t^i, \mathbf{x}_{t-1}^i, \mathbf{y}_t\right)$
5. (Sample) Draw $\theta_t^i \sim p(\theta_t \mid \mathbf{s}_t^i)$
6. (Update) $(\mathbf{s}_t^x)^i = \mathcal{K}\left((\mathbf{s}_{t-1}^x)^{k^i}, \theta_t^i, \mathbf{y}_t\right)$

Where $\mathcal{S}$ denotes the deterministic updating function for parameter sufficient statistics and $\mathcal{K}$ denotes Kalman filter for state sufficient statistics update.

3. Statistical Factor Analysis

Statistical factor analysis is a technique used to explain the correlation structure of multivariate observations through a lower-dimensional set of unobserved factors. The model can be expressed as:

$$\mathbf{y}_t = \mathbf{B}\mathbf{x}_t + \mathbf{v}_t \tag{29}$$

where $\mathbf{y}_t$ is $(s \times 1)$ vector of observations, $\mathbf{x}_t$ is $(k \times 1)$ vector of statistical factors, and $\mathbf{B}$ is $(s \times k)$ factor loading matrix. $\mathbf{v}_t$ is the $(s \times 1)$ error term that follows certain distribution assumption. The hidden factors $\mathbf{x}_t$ are often assumed to follow an i.i.d. standard normal distribution, i.e. $\mathbf{x}_t \sim N(0, \mathbf{I}_k)$, and are assumed to be independent of the error term $\mathbf{v}_t$.

3.1. MLE Factor Analsis

The most commonly used approach for estimating the factor model in equation (29) is the Expectation-Maximization (EM) algorithm for maximum likelihood estimation (MLE) [4–6]. The EM algorithm for maximum likelihood estimation of the factor model is outlined below as Algorithm 9:

Algorithm 9 EM algorithm for Statistical Factor Analysis

1. E-Step: Given current fit of $\mathbf{B}$ and $\mathbf{R}$, calculate sufficient statistics:

$$\mathbf{fl} = \mathbf{B}^T\left(\mathbf{B}\mathbf{B}^T + \mathbf{R}\right)^{-1} \tag{30}$$

$$E[\mathbf{x}_t \mid \mathbf{y}_t] = \mathbf{fl}\mathbf{y}_t \tag{31}$$

$$E\left[\mathbf{x}_t\mathbf{x}_t^T \mid \mathbf{y}_t\right] = \mathbf{I} - \mathbf{fl}\mathbf{B} + \mathbf{fl}\mathbf{y}_t\mathbf{y}_t^T\mathbf{fl} \tag{32}$$

2. M-Step: Update parameter estimation,

$$\hat{\mathbf{B}} = \left(\sum_{t=1}^T \mathbf{y}_t E[\mathbf{x}_t \mid \mathbf{y}_t]\right)\left(\sum_{t=1}^T E\left[\mathbf{x}_t\mathbf{x}_t^T \mid \mathbf{y}_t\right]\right)^{-1} \tag{33}$$

$$\hat{\mathbf{R}} = \frac{1}{T}\mathrm{diag}\left(\sum_{t=1}^T \left(\mathbf{y}_t\mathbf{y}_t^T - \hat{\mathbf{B}}E[\mathbf{x}_t \mid \mathbf{y}_t]\mathbf{x}_t^T\right)\right) \tag{34}$$

 doi:10.20944/preprints202507.2022.v1

The EM algorithm starts with certain initial values for $\mathbf{B}$ and $\mathbf{R}$ and terminates when the rate of change in the log likelihood falls below some critical value. The log likelihood can be computed as:

$$\begin{aligned}\log L(\mathbf{B},\mathbf{R}\mid \mathbf{y}_{1:T}) &= -\frac{1}{2}\Bigg(T\Big(s\log(2\pi)+\log\Big(\det\Big(\mathbf{B}\mathbf{B}^T+\mathbf{R}\Big)\Big)\Big)\\ &\quad+\sum_{t=1}^{T}\mathbf{y}_t^T\Big(\mathbf{B}\mathbf{B}^T+\mathbf{R}\Big)^{-1}\mathbf{y}_t\Bigg)\end{aligned} \tag{35}$$

3.2. Mixture of Factor Analyzer

The factor model in equation (29) is a global linear model. However, in many real world applications — such as in financial engineering — it fails to reflect the fact that the observed data present multiple regimes whose behaviors differ materially. To address this limitation, the Mixture of Factor Analyzer (MFA) model extends the MLE statistical factor model by combining local dimension reduction method with a Gaussian mixture model for clustering. Specifically, the MFA model can be described in the following form:

$$\mathbf{y}_t = \mathbf{B}(i)\mathbf{x}_t + \mathbf{v}_t \text{ with prob. } \pi_i \ (i=1,\ldots,m) \tag{36}$$

where $\sum_{i=1}^{m}\pi_i = 1$, $\pi_i \geq 0$, and the index i denotes the component (or regime), drawn from a categorical distribution with mixing probabilities $\{\pi_i\}$. $\mathbf{B}(i)$ is the factor loading matrix specific to component i. Following the formulation in [6], we assume the observation covariance $\mathbf{R}$ for each component are the same, and the number of mixture is fixed prior to the estimation step. The EM algorithm for estimating the parameters of the MFA model is described below in Algorithm 10:

Algorithm 10 EM algorithm for Mixture of Factor Analyzer

1. E-Step: Given current fit of π_i for $i=1,\ldots,m$, $\mathbf{B}$ and $\mathbf{R}$, calculate,

$$\tau_{ij} = \frac{\pi_i N\Big(\mathbf{y}_j, \mathbf{B}(i)\mathbf{B}(i)^T+\mathbf{R}\Big)}{\sum_{i=1}^{m}\pi_i N\Big(\mathbf{y}_j, \mathbf{B}(i)\mathbf{B}(i)^T+\mathbf{R}\Big)} \tag{37}$$

$$\gamma(i) = \mathbf{B}(i)^T\Big(\mathbf{B}(i)\mathbf{B}(i)^T+\mathbf{R}\Big)^{-1} \tag{38}$$

$$\mathbf{\Sigma}(i) = \frac{1}{T}\sum_{j=1}^{T}\Big(\sqrt{\tau_{ij}}\mathbf{y}_j\Big)\Big(\sqrt{\tau_{ij}}\mathbf{y}_j\Big)^T \tag{39}$$

2. M-Step:

$$\hat{\mathbf{B}}(i) = \frac{\mathbf{\Sigma}(i)\gamma(i)^T}{\frac{1}{T}\sum_{j=1}^{T}\boldsymbol{\tau}_{ij}(\mathbf{I}_s-\gamma(i)\mathbf{B}(i))+\gamma(i)\mathbf{\Sigma}(i)\gamma(i)^T} \tag{40}$$

$$\hat{\mathbf{R}} = \frac{1}{T}\mathrm{diag}\Bigg(\sum_{i=1}^{m}\mathbf{\Sigma}(i)-\sum_{i=1}^{m}\hat{\mathbf{B}}(i)\gamma(i)\mathbf{\Sigma}(i)\Bigg) \tag{41}$$

$$\hat{\pi}_i = \frac{1}{T}\sum_{j=1}^{T}\boldsymbol{\tau}_{ij} \tag{42}$$

Starting with initial values of $\mathbf{ß}_i$ for $i=1,\ldots,m$, $\mathbf{B}(i)$, and $\mathbf{R}$, we can iterate algorithm 10 until convergence. The log-likelihood of the observed data under the MFA model is given by:

$$\log L(\mathbf{\Psi}) = \sum_{j=1}^{T}\log\Bigg\{\sum_{i=1}^{m}\pi_i N\Big(\mathbf{y}_j, \mathbf{B}(i)\mathbf{B}(i)^T+\mathbf{R}\Big)\Bigg\}$$

 doi:10.20944/preprints202507.2022.v1

where $\mathbf{\Psi}$ denotes the set of all model parameters.

3.3. Regime Switching Dynamic Factor Model

Although the mixture of factor analyzer (MFA) model can capture data generated from multiple regimes, it still assumes the observations are i.i.d. and fails to account for the temporal persistence of market regimes. In practice, particularly in financial applications, market regimes tend to persist over time, with each regime exhibiting distinct statistical properties. Regime switching model equipped with Markov chain mechanism can better model the regime persistence and capture the abrupt regime shifts more efficiently than commonly used "low-pass filtering" approaches such as rolling window estimation. It fits better for financial time series, where capturing time dynamics is essential.

3.3.1. Methods

In this work, we represent the regime switching dynamic factor model within a state space framework. The estimation of regime switching state space models has been well studied in the literature [8,24,25], with most approaches relying on the Markov Chain Monte Carlo (MCMC) algorithms. MCMC approaches are not only computationally very expensive, but also suffer slow convergence issue. Moreover, they are batch (offline) algorithms in nature which limits their use in real-time trading.

In this work, we will rely on the particle learning technique that is introduced in section 2.4 as our estimation method. Computationally, it's significantly faster than MCMC algorithms and, being an online algorithm, it can be adopted in the real time trading environment. The convergence of particle filtering algorithm is well studied in the literature [17,26].

Another key advantage of the particle filtering framework is its modular design, which makes it highly adaptable to complex model structures. Additional structure can be incorporated into the system relatively easier than MCMC framework, as long as it remains possible to draw samples from the importance density and evaluate the likelihood function upon receiving the new observations (See [14,27] for examples).

To introduce our model, we first define a discrete latent regime variable $r_t \in \{1, \ldots, K\}$ which captures the regime at time t. The essential state used in the particle learning algorithm is then extended as:

$$\mathbf{Z}_t = \{r_t, \mathbf{s}_t, \mathbf{s}_t^x, \theta\},$$

where $\mathbf{s}_t$ and $\mathbf{s}_t^x$ denote the sufficient statistics for the parameters and the hidden states, respectively, and θ denotes the model parameters.

The model is specified as:

$$\mathbf{x}_t = \mathbf{A}\mathbf{x}_t + \mathbf{w}_t \tag{43}$$

$$\mathbf{y}_t = B(r_t)\mathbf{x}_t + \sqrt{\lambda_t}\mathbf{v}_t \tag{44}$$

$$\lambda_t \sim IG(v/2, v/2) \tag{45}$$

where $IG(\cdot)$ denotes the inverse Gamma distribution, $\mathbf{v}_t \sim N(0_s, \mathbf{R}_s)$, $\mathbf{w}_t \sim N(0_k, \mathbf{I}_k)$. In this representation, we model heavy tailed innovation using the data augmentation scheme of [28], where at each time t, we will follow the two step approach: i) $\lambda_t \sim$ Inv-Gamma$(v/2, v/2)$, ii) $\mathbf{\jmath}_t = \sqrt{\lambda_t}\mathbf{v}_t$, so that $\mathbf{\jmath}_t \sim t_v(0_s, \mathbf{R}_s)$ where $\mathbf{R}_s$ is a diagonal matrix.

If the degree of freedom is given, the at each time step t, our system is a conditionally Gaussian model. To derive the particle learning algorithm for this model (equation (43) to (45)), we compute the predictive density $p(\mathbf{y}_t \mid \mathbf{z}_{t-1}, \lambda_t)$ by integrating out the latent regime variable r_t:

$$\begin{aligned} p(\mathbf{y}_t \mid \mathbf{z}_{t-1}, \lambda_t) &= \sum_{r_t=1}^{K} p\left(\mathbf{y}_t \mid r_t, \left(\mathbf{s}_{t-1}^x, \theta, \lambda_t\right)^i\right) \\ &\quad \cdot p(r_t \mid r_{t-1}, \theta) \end{aligned} \tag{46}$$

 doi:10.20944/preprints202507.2022.v1

where $p\left(\mathbf{y}_t \mid r_t, \left(\mathbf{s}_{t-1}^x, \theta, \lambda_t\right)^i\right)$ is the conditional likelihood, which is derived in Appendix A, and $p(r_t \mid r_{t-1}, \theta)$ is given by the estimated transition matrix. To propagate r_t, we sample from the posterior distribution:

$$p\left(r_t \mid \left(r_{t-1}, \mathbf{s}_{t-1}^x, \theta, \lambda_t\right)^i, \mathbf{y}_t\right) \propto p\left(\mathbf{y}_t \mid r_t, \left(\mathbf{s}_{t-1}^x, \theta, \lambda_t\right)^i\right) \cdot p(r_t \mid r_{t-1}, \theta) \tag{47}$$

Algorithm 11 PL for Regime Switching Dynamic Factor Model

1. (Sampling) Draw $\lambda_t^i \sim IG(v/2, v/2)$ ($\lambda_t^i = 1$ for Gaussian model)
2. (Resample) Draw $k^i \sim Multi\{W_t^1, \ldots, W_t^N\}$, with $W_t^i \propto p\left(\mathbf{y}_t \mid \mathbf{z}_{t-1}^i, \lambda_t\right)$ where

$$p\left(\mathbf{y}_t \mid \mathbf{z}_{t-1}^i, \lambda_t\right) = \sum_{k=1}^{K} N\left(\mathbf{y}_t; \mathbf{m}_y(k), \mathbf{C}_y(k)\right) p\left(r_t = k \mid r_{t-1}^i, \boldsymbol{\theta}\right)$$

   and $\mathbf{m}_y(k) = \mathbf{B}(k)\mathbf{A}\mathbf{m}_{t-1}$, $\mathbf{C}_y(k) = \mathbf{B}(k)\mathbf{A}\mathbf{C}_{t-1}\mathbf{A}^T\mathbf{B}(k)^T + \mathbf{B}(k)\mathbf{B}(k)^T + \lambda_t\mathbf{R}_s$.
3. (Propagate) Draw $r_t^i \sim p\left(r_t \mid \left(\mathbf{s}_{t-1}^x, r_{t-1}, \boldsymbol{\theta}\right)^{k^i}, \mathbf{y}_t\right)$, which follows multinomial distribution with:

$$p\left(r_t = k \mid \mathbf{s}_{t-1}^x, r_{t-1}, \boldsymbol{\theta}, \mathbf{y}_t\right) \propto N\left(\mathbf{y}_t; \mathbf{m}_y^{k^i}(k), \mathbf{C}_y^{k^i}(k)\right) p\left(r_t = k \mid r_{t-1}^{k^i}, \boldsymbol{\theta}^{k^i}\right)$$

4. (Propagate) Draw $\mathbf{x}_t^i \sim p\left(\mathbf{x}_t \mid r_t^i, \mathbf{x}_{t-1}^{k^i}, \boldsymbol{\theta}^{k^i}, \mathbf{y}_t\right)$ from normal distribution $N(\mathbf{m}_x, \mathbf{C}_x)$ with:

$$\mathbf{m}_x = \mathbf{C}_x\left(\mathbf{B}(r_t)^T(\lambda_t\mathbf{R}_s)^{-1}\mathbf{y}_t + \mathbf{A}\mathbf{x}_{t-1}\right) \tag{48}$$

$$\mathbf{C}_x^{-1} = \mathbf{B}(r_t)^T(\lambda_t\mathbf{R}_s)^{-1}\mathbf{B}(r_t) + \mathbf{I}_k \tag{49}$$

5. (Update) $\mathbf{s}_t^i = \mathbf{S}\left(\mathbf{s}_{t-1}^{k^i}, r_t^i, r_{t-1}^{k^i}, \mathbf{x}_t^i, \mathbf{x}_{t-1}^{k^i}, \mathbf{y}_t\right)$.
6. (Sample) Draw $\boldsymbol{\theta}^i \sim p\left(\boldsymbol{\theta} \mid \mathbf{s}_t^i\right)$.
7. (Update) $\left(\mathbf{s}_t^x\right)^i = \mathbf{K}\left(\left(\mathbf{s}_{t-1}^x\right)^{k^i}, \boldsymbol{\theta}^i, \mathbf{y}_t\right)$ and draw estimated states $\hat{\mathbf{x}}_t^i \sim N\left(\mathbf{m}_{x1}^i, \mathbf{C}_{x1}^i\right)$.

where the deterministic updating formula $\mathbf{S}$ in step 5 is given by:

$$v_t = \varphi^2 v_{t-1} + 1 \tag{50}$$

$$\mathbf{g}_t = \boldsymbol{\Phi}_{t-1}\mathbf{x}_t \tag{51}$$

$$\zeta_t^2 = \lambda_t\varphi^2 + \mathbf{x}_t^T\mathbf{g}_t \tag{52}$$

$$\hat{\mathbf{e}}_t = \mathbf{y}_t - \mathbf{b}_{t-1}^T\mathbf{x}_t \tag{53}$$

$$\mathbf{b}_t = \mathbf{b}_{t-1} + \frac{1}{\zeta_t^2}\mathbf{g}_t\hat{\mathbf{e}}_t^T I_{r_t=k} \tag{54}$$

$$\boldsymbol{\Phi}_t = \frac{1}{\varphi^2}\left(\boldsymbol{\Phi}_{t-1} - \frac{1}{\zeta_t^2}\mathbf{g}_t\mathbf{g}_t^T I_{r_t=k}\right) \tag{55}$$

$$\mathbf{V}_t = \varphi^2\left(\mathbf{V}_{t-1} + \sum_{k=1}^{K}\frac{1}{\zeta_t^2}\hat{\mathbf{e}}_t\hat{\mathbf{e}}_t^T I_{r_t=k}\right) \tag{56}$$

$$\mathbf{g}_t^x = \boldsymbol{\Psi}_{t-1}\mathbf{x}_{t-1} \tag{57}$$

$$\sigma_t^2 = \varphi^2 + \mathbf{x}_{t-1}^T\mathbf{g}_t^x \tag{58}$$

$$\hat{\mathbf{e}}_t^x = \mathbf{x}_t - \mathbf{a}_{t-1}^T\mathbf{x}_{t-1} \tag{59}$$

$$\mathbf{a}_t = \mathbf{a}_{t-1} + \frac{1}{\sigma_t^2}\mathbf{g}_t^x\hat{\mathbf{e}}_t^{xT} \tag{60}$$

$$\boldsymbol{\Psi}_t = \frac{1}{\varphi^2}\left(\boldsymbol{\Psi}_{t-1} - \frac{1}{\sigma_t^2}\mathbf{g}_t^x\mathbf{g}_t^{xT}\right) \tag{61}$$

 doi:10.20944/preprints202507.2022.v1

where $i = 1, \ldots, s$. The sampling density in step 4 is derived in Appendix B. As discussed in Appendix C, we can assume each row of transition matrix $\mathbf{T}$, $\mathbf{p}_i$, follows a Dirichlet distribution $Dir(\mathbf{p}_i \mid \mathbf{ff}_i)$, for $i = 1, \ldots, K$, and their sufficient statistics can be updated as:

$$\alpha_{i,j,t} = \alpha_{i,j,t-1} + I_{r_t=j, r_{t-1}=i}, \text{ for } i, j = 1, \ldots, K \tag{62}$$

For step 6, we can either simulate model parameters from the updated sufficient statistics or calculate them analytically. The analytical solution is give by:

$$\begin{aligned} \mathbf{B}(r_t) &= \mathbf{b}_{k,t} && (63)\\ \mathbf{R}_s &= \mathbf{V}_t/(v_t - 2) && (64)\\ \mathbf{A} &= \mathbf{a}_t && (65)\\ \mathbf{T}_{ij} &= \frac{\alpha_{i,j,t} - 1}{\alpha_{0,t} - K} && (66) \end{aligned}$$

where we use the posterior mode (MAP) as transition probability estimate.

For step 7, $\mathbf{K}$ stands for Kalman filter, and the specific algorithm is given by 2.

3.3.2. Simulation Analysis

In this section, we apply our particle learning algorithm to synthetic data to validate the estimation algorithm. We generate synthetic scenarios based on a three regime dynamic factor model involving 15 synthetic instruments. Our "true" factor loading matrices, observation covariance and "true" state transition matrix are specified as follows:

$$\mathbf{B}(r_t = 1) = \begin{bmatrix} 0.93 & 0.316 & 0.184 \\ 0.205 & 0.568 & 0.596 \\ \vdots & \vdots & \vdots \\ 0.81 & 0.096 & 0.219 \end{bmatrix}$$

$$\mathbf{B}(r_t = 2) = \begin{bmatrix} -0.56 & 1.011 & 0.945 \\ 2.821 & -1.165 & 1.486 \\ \vdots & \vdots & \vdots \\ 0.152 & 2.381 & -0.153 \end{bmatrix}$$

$$\mathbf{B}(r_t = 3) = \begin{bmatrix} -0.694 & -0.654 & -0.443 \\ -0.15 & -2.678 & -0.268 \\ \vdots & \vdots & \vdots \\ -0.917 & -0.849 & -0.811 \end{bmatrix}$$

$$\text{Diag}\mathbf{R} = \begin{bmatrix} 0.49 & 0.211 & \cdots & 0.518 & 0.488 \end{bmatrix}$$

$$\mathbf{A} = \begin{bmatrix} 0.553 & 0 & 0 \\ 0 & 0.477 & 0 \\ 0 & 0 & 0.312 \end{bmatrix}$$

and regime transition matrix $\mathbf{T}$ is:

$$\mathbf{T} = \begin{bmatrix} 0.995 & 0.0025 & 0.0025 \\ 0.0025 & 0.995 & 0.0025 \\ 0.0025 & 0.0025 & 0.995 \end{bmatrix}$$

 doi:10.20944/preprints202507.2022.v1

Since factor loadings and factor returns are not separately identifiable in our model structure, in this analysis, we mainly check regimes detection (See Figure 1), idiosyncratic risk estimation (See Figure 2), and cumulative expected returns' tracking (See Figure 3). The estimation results are as follows:

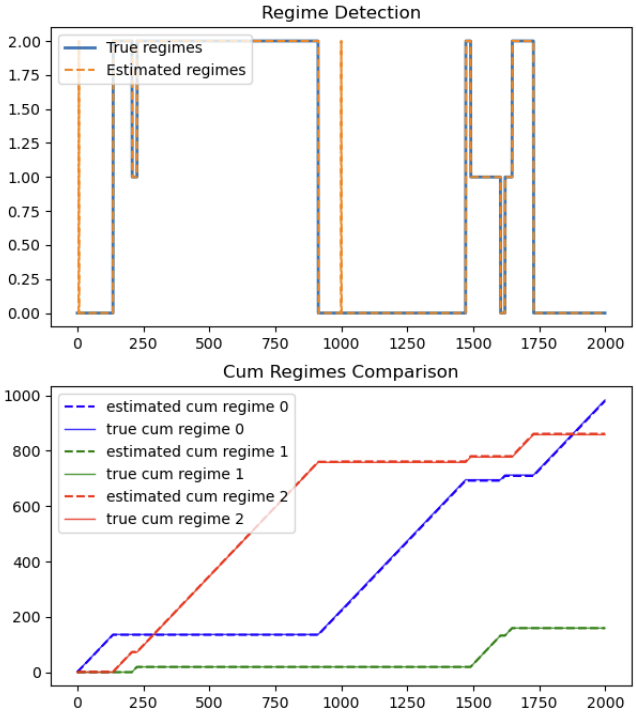


Figure 1. Regime Detection.

As shown in Figure 1, the algorithm accurately captures the evolution of hidden regimes from both regime detection plot and cumulative regimes comparison plot.

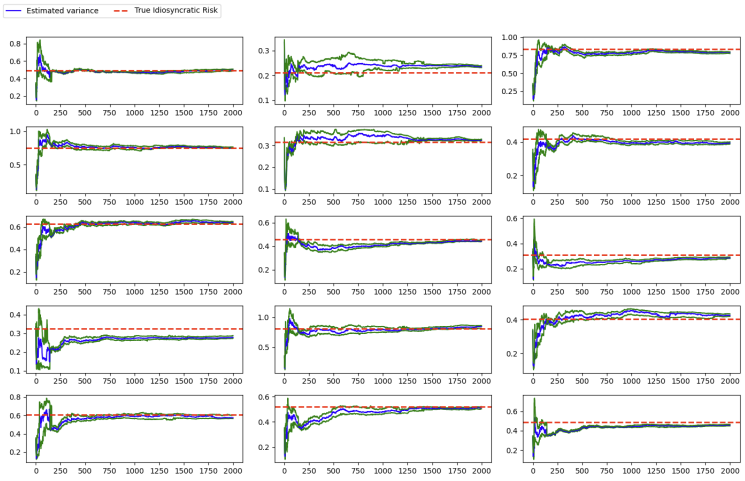


Figure 2. Idiosyncratic Risk Estimation.

In terms if idiosyncratic risk (or error variance), the algorithm can also converge to their unbiased estimation.

 doi:10.20944/preprints202507.2022.v1

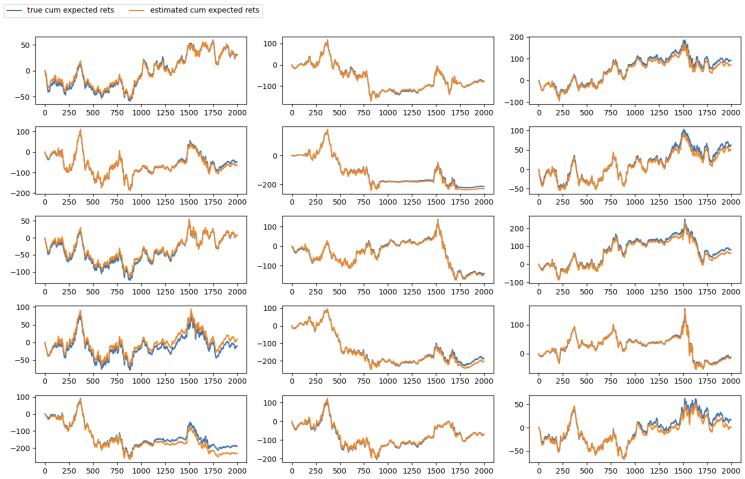


Figure 3. Cumulative Expected Return Tracking.

The most important metric in statistical factor analysis is how accurately the model tracks the expected return. In this synthetic setting, our model demonstrates strong performance in the expected return (or cumulative expected return as shown in Figure 3) tracking.

4. Results in Statistical Arbitrage Strategy

In both academic and industry, equity statistical arbitrage strategy (equity stat-arb) usually means the idea of trading groups of stocks against either explicit or synthetic factors, which can be seen as the generalization of "pairs-trading" (see [29]). In some cases, we would long an individual stock and short the factors, and in others we would short the stock and long the factors. Overall, due to the netting of long and short positions, our exposure to the factors will be small. One key step of the equity stat-arb strategy is the decomposition of assets' returns into the systematic part and idiosyncratic part. When we use a more effective factor model, we can have a better decomposition and hence the better strategy performance.

4.1. Market Neutral Strategy

In this paper, we evaluate the quality of factor models within the context of an equity statistical arbitrage strategy. Inspired by the approach of [30], in order to avoid the unnecessary complexity of signal generation and potential overfitting issues we adopt a deliberately simple reversion signal which is the negative value of previous day's residuals. To ensure market neutrality, we construct a dollar-neutral portfolio each day, with equal dollar amounts allocated to long and short positions.

To be specific, we compare the performance of trading strategies derived from both regime switching dynamic factor model and static MLE factor model. For the regime switching dynamic factor model, the factor loading matrices for each regime are initialized using estimates from a mixture of factor analyzer (MFA), trained on a "burnin" data from 2005-01-03 to 2011-01-01. The residualization step is implemented in a rolling window framework so that the residuals are always generated out-of-sample.

After the residualization step, we can generate our signals using naive rule-based approach. If previous day's residual for asset i has $v_{i,t-1} > d$, where d is the threshold, then we will say that the asset i is over-priced at time $t-1$ and short this asset for the next time stamp t. For $v_{i,t-1} < -d$, we will do the opposite operation. At each time step, we form a dollar-neutral portfolio with equally weighted positions on both the long and short sides.

4.2. Performance Comparison

We run this naive strategy using both dynamic factor model and static factor model respectively, and the strategy performance is given as Figure 4:

 doi:10.20944/preprints202507.2022.v1

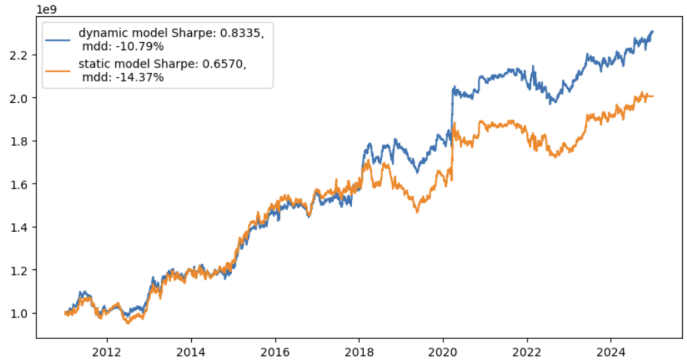
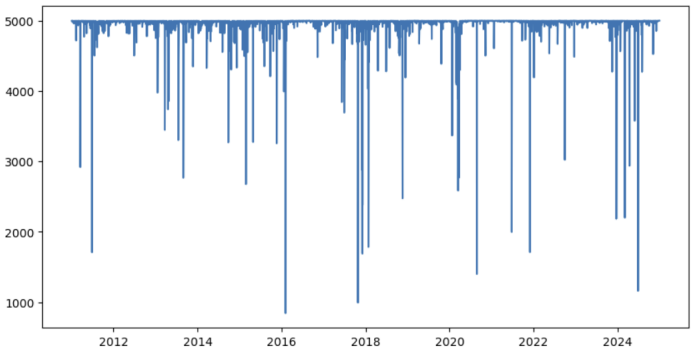
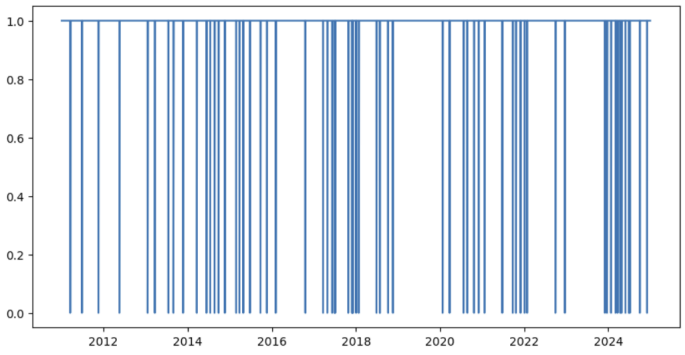


Figure 4. Equity Stat-Arb Performance Comparison.

We can see from Figure 4 that the strategy has a better Sharpe and lower Maximum drawdown when using regime switching dynamic factor model than using static MLE factor model. The associated historical effective sample size plot and regime detection plot are given in Figure 5:



(a)



(b)

Figure 5. (a) Effective Sample Size at each day when running particle filter (a) Regime detection when using regime switching particle filter.

From the effective sample size, we can see the particle filter estimation is stable since most of the sample size are larger than 1000 for a three-factor model. From the regimes detection, we can see there are more regime changes in 2024 and indeed the strategy relying on regime switching dynamic factor model performs better in 2024 than the one using static model.

5. Discussion

This paper investigates the estimation of regime switching dynamic factor model using a particle learning algorithm. Through simulation studies, we validate our estimation approach by evaluating regime detection accuracy, idiosyncratic risk estimation, and the model's ability to track expected returns. Empirical study in the equity statistical arbitrage framework further demonstrates that

 doi:10.20944/preprints202507.2022.v1

the regime-switching dynamic factor model outperforms a conventional static MLE factor model in capturing underlying data dynamics.

The particle learning algorithm implemented in this paper builds on the framework of [15], with several modifications. First, we extend the innovation distribution to a mixture of Gaussian distributions. Second, we simplify the parameter learning step by only tracking the sufficient statistics, rather than using the marginalization via simulation. Last, we apply the estimated model to the empirical data and compare its performance against the conventional MLE factor model.

We are able to see the performance improvements in the equity statistical arbitrage strategy when using our model. Our model not only improves the Sharpe ratio but also minimizes the maximum drawdown. These results align with our initial motivation: It is well known that financial market operates under multiple regimes, incorporating the regime switching mechanism that is governed by hidden Markov model in our model representation allows us better model the structural shifts and enhances risk management — particularly during periods of elevated volatility and contagion risk.

While the current particle learning algorithm is based on the vanilla auxiliary particle filter, there remains a lot of room to improve the quality of our particle filtering algorithm and hence improve the tracking of hidden states (or hidden factor in this model). For instance, the ABC-based sequential Monte Carlo filter [31] could further enhance the filtering accuracy. Additionally, incorporating more robust, heavy-tailed innovation distributions [32] may improve the robustness of our estimation, particularly in the presence of extreme market movements.

Appendix A. Probability Density Function of $p(\mathbf{y}_t \mid \mathbf{z}_{t-1})$

The likelihood function $p(\mathbf{y}_t \mid \mathbf{z}_{t-1})$ is straightforward to derive. Given $\mathbf{z}_{t-1} = \{\mathbf{s}_{t-1}, \mathbf{m}_{t-1}, \mathbf{C}_{t-1}, \theta_{t-1}\}$, and the equation $\mathbf{y}_t = \mathbf{B}(\mathbf{A}\mathbf{x}_{t-1} + \mathbf{w}_t) + \mathbf{v}_t$, the likelihood function $p(\mathbf{y}_t \mid \mathbf{z}_{t-1})$ follows the Normal distribution $N(\mathbf{m}_y, \mathbf{C}_y)$, where

$$\begin{aligned}\mathbf{m}_y &= \mathbf{B}\mathbf{A}\mathbf{m}_{t-1} \\ \mathbf{C}_y &= \mathbf{B}\Big(\mathbf{A}\mathbf{C}_{t-1}\mathbf{A}^T + \mathbf{I}_k\Big)\mathbf{B}^T + \mathbf{R}_s\end{aligned}$$

Appendix B. Probability Density Function of $p(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \theta_{t-1}, \mathbf{y}_t)$

Based on Bayes' rule, we can factor $p(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \boldsymbol{\theta}_{t-1}, \mathbf{y}_t)$ as:

$$\begin{aligned}p(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \boldsymbol{\theta}_{t-1}, \mathbf{y}_t) &\propto p(\mathbf{y}_t \mid \mathbf{x}_t, \boldsymbol{\theta}_{t-1}) p(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \boldsymbol{\theta}_{t-1}) \\ &\propto \exp\left\{-\frac{1}{2}(\mathbf{y}_t - \mathbf{B}\mathbf{x}_t)^T \mathbf{R}_s^{-1}(\mathbf{y}_t - \mathbf{B}\mathbf{x}_t)\right\} \\ &\quad \exp\left\{-\frac{1}{2}(\mathbf{x}_t - \mathbf{A}\mathbf{x}_{t-1})^T(\mathbf{x}_t - \mathbf{A}\mathbf{x}_{t-1})\right\}\end{aligned}$$

Expressing quadratic form to normal form:

$$\begin{aligned}&(\mathbf{y}_t - \mathbf{B}\mathbf{x}_t)^T \mathbf{R}_s^{-1}(\mathbf{y}_t - \mathbf{B}\mathbf{x}_t) + (\mathbf{x}_t - \mathbf{A}\mathbf{x}_{t-1})^T(\mathbf{x}_t - \mathbf{A}\mathbf{x}_{t-1}) \\ &\qquad = \begin{bmatrix} \bar{\mathbf{U}}\mathbf{y}_t - \bar{\mathbf{U}}\mathbf{B}\mathbf{x}_t \\ \mathbf{A}\mathbf{x}_{t-1} - \mathbf{x}_t \end{bmatrix}^T \begin{bmatrix} \bar{\mathbf{U}}\mathbf{y}_t - \bar{\mathbf{U}}\mathbf{B}\mathbf{x}_t \\ \mathbf{A}\mathbf{x}_{t-1} - \mathbf{x}_t \end{bmatrix}\end{aligned}$$

where $\bar{\mathbf{U}}^T\bar{\mathbf{U}} = \mathbf{R}_s^{-1}$. If we let $\bar{\mathbf{V}} = \begin{bmatrix} \bar{\mathbf{U}}\mathbf{y}_t \\ \mathbf{A}\mathbf{x}_{t-1} \end{bmatrix}$, $\bar{\mathbf{W}} = \begin{bmatrix} \bar{\mathbf{U}}\mathbf{B} \\ \mathbf{I}_k \end{bmatrix}$, then

$$\begin{aligned}(\bar{\mathbf{V}} - \bar{\mathbf{W}}\mathbf{x}_t)^T(\bar{\mathbf{V}} - \bar{\mathbf{W}}\mathbf{x}_t) &= (\bar{\mathbf{V}} - \bar{\mathbf{W}}\hat{\mathbf{x}}_t)^T(\bar{\mathbf{V}} - \bar{\mathbf{W}}\hat{\mathbf{x}}_t) \\ &\quad + (\mathbf{x}_t - \hat{\mathbf{x}}_t)^T\bar{\mathbf{W}}^T\bar{\mathbf{W}}(\mathbf{x}_t - \hat{\mathbf{x}}_t)\end{aligned}$$

 doi:10.20944/preprints202507.2022.v1

where $\hat{\mathbf{x}}_t = \left(\bar{\mathbf{W}}^T\bar{\mathbf{W}}\right)^{-1}\bar{\mathbf{W}}^T\bar{\mathbf{V}}$. Hence,

$$\begin{aligned}\mathbf{m}_x &= \mathbf{C}_x\left(\mathbf{B}^T\mathbf{R}_s^{-1}\mathbf{y}_t + \mathbf{A}\mathbf{x}_{t-1}\right)\\ \mathbf{C}_x^{-1} &= \mathbf{B}^T\mathbf{R}_s^{-1}\mathbf{B} + \mathbf{I}_k\end{aligned}$$

Appendix C. Conjugate Prior of Categorical Distribution

The Dirichlet distribution, which is a multivariate generalization of the Beta distribution, is usually used as the conjugate prior distribution of the categorical distribution (also known as multinomial distribution) (See [33]). The probability density function of the Dirichlet distribution is defined as:

$$Dir(\mathbf{x} \mid \boldsymbol{\alpha}) = \frac{1}{B(\boldsymbol{\alpha})}\prod_{i=1}^{K} x_i^{\alpha_i - 1}$$

where $\mathbf{x} = \{x_1, \ldots, x_K\}$ lives in a K ($K \geq 2$) dimensional probability simplex with constraints $\sum_{i=1}^{K} x_i = 1$, and $x_i > 0$ for all $i \in [1, K]$. $\boldsymbol{\alpha} = \{\alpha_1, \ldots, \alpha_K\}$ is called concentration parameters with constraint $\alpha_i > 0$ for all $i \in [1, K]$. Assume we have a set of data $\mathbf{y} = \{y_1, \ldots, y_T\}$ generated from Categorical distribution with parameters $\mathbf{p} = \{p_1, \ldots, p_K\}$, $\sum_{i=1}^{K} p_i = 1$ and $p_i > 0$. Then we can represent likelihood as:

$$p(\mathbf{y} \mid \mathbf{p}) = \prod_{i=1}^{K} p_i^{N_i}$$

where $N_i = \sum_{t=1}^{T} \delta(y_t = i)$, and $\delta(\cdot)$ is the Dirac Delta function. Since the parameter vector $\mathbf{p}$ also lives in a K dimensional probability simplex, and the Dirichlet distribution belongs to exponential family, so it becomes a natural choice of conjugate prior for parameter $\mathbf{p}$. If we assume prior distribution for $\mathbf{p}$ as:

$$Dir(\mathbf{p} \mid \boldsymbol{\alpha}) = \frac{1}{B(\boldsymbol{\alpha})}\prod_{i=1}^{K} p_i^{\alpha_i - 1}$$

Hence, the posterior is also Dirichlet:

$$\begin{aligned}p(\mathbf{p} \mid \mathbf{y}) &\propto Dir(\mathbf{p} \mid \boldsymbol{\alpha})p(\mathbf{y} \mid \mathbf{p})\\ &\propto \prod_{k=1}^{K} p_i^{N_i} p_i^{\alpha_i - 1}\\ &\propto \prod_{k=1}^{K} p_i^{\alpha_i + N_i - 1}\\ &= Dir(\mathbf{p} \mid \alpha_1 + N_1, \ldots, \alpha_K + N_K)\end{aligned}$$

In other words, the posterior sufficient statistics $\tilde{\alpha}_i$ can be updated by simply adding the empirical counts N_i. It is easy to show the posterior mode (MAP estimate) is given by:

$$\tilde{p}_i = \frac{\alpha_i + N_i - 1}{\alpha_0 + N - K}$$

where $\alpha_0 = \sum_{i=1}^{K} \alpha_i$, $N = \sum_{i=1}^{K} N_i$. At the meanwhile, the posterior mean is given by:

$$\hat{p}_i = \frac{\alpha_i + N_i}{\alpha_0 + N}$$

References

1. Barra, M. *Barra Risk Model Handbook*; MSCI, 2004.
2. Jolliffe, I.T. *Principal component analysis*, 2. ed., [nachdr.] ed.; Springer series in statistics, Springer: New York, 2004.

 doi:10.20944/preprints202507.2022.v1

3. Hinton, G.; Dayan, P.; Revow, M. Modeling the manifolds of images of handwritten digits. *IEEE Transactions on Neural Networks* **1997**, *8*, 65–74. https://doi.org/10.1109/72.554192.
4. McLachlan, G.; Peel, D. *Finite Mixture Models*; Wiley, 2000. https://doi.org/10.1002/0471721182.
5. Rubin, D.B.; Thayer, D.T. EM Algorithms for ML Factor Analysis. *Psychometrika* **1982**, *47*, 69–76. https://doi.org/10.1007/bf02293851.
6. Zoubin Ghahramani, G.E.H. Parameter Estimation for Linear Dynamical Systems. *Technical Report CRG-TR-96-2* **1996**.
7. Bai, J.; Wang, P. Identification and Bayesian Estimation of Dynamic Factor Models. *Journal of Business Economic Statistics* **2014**, *33*, 221–240. https://doi.org/10.1080/07350015.2014.941467.
8. Kim, C.J.; Halbert, D.C.R. *State-Space Models with Regime Switching: Classical and Gibbs-Sampling Approaches with Applications*; The MIT Press, 1999. https://doi.org/10.7551/mitpress/6444.001.0001.
9. Geweke, J.; Zhou, G. Measuring the price of the Arbitrage Pricing Theory. *The Review of Financial Stuidies, vol. 9, no.2* **1996**.
10. Forni, Mario, D.G.M.L.; Reichlin, L. Opening the Black Box: Structure Factor Models with Large Cross Sections. *Econometric Theory 25, no. 5* **2009**.
11. Doucet, A.; Johansen, A.M. A tutorial on particle filtering and smoothing: Fifteen years later. *Handbook of nonlinear filtering* **2009**.
12. Arulampalam, M.; Maskell, S.; Gordon, N.; Clapp, T. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing* **2002**, *50*, 174–188. https://doi.org/10.1109/78.978374.
13. Liu, J.; West, M., Combined Parameter and State Estimation in Simulation-Based Filtering. In *Sequential Monte Carlo Methods in Practice*; Springer New York, 2001; pp. 197–223. https://doi.org/10.1007/978-1-4757-3437-9_10.
14. Djuric, P.M.; Khan, M.; Johnston, D.E. Particle Filtering of Stochastic Volatility Modeled With Leverage. *IEEE Journal of Selected Topics in Signal Processing* **2012**, *6*, 327–336. https://doi.org/10.1109/jstsp.2012.2201695.
15. Carvalho, C.M.; Johannes, M.S.; Lopes, H.F.; Polson, N.G. Particle Learning and Smoothing. *Statistical Science* **2010**, *25*. https://doi.org/10.1214/10-sts325.
16. Gordon, N.; Salmond, D.; Smith, A. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings F Radar and Signal Processing* **1993**, *140*, 107. https://doi.org/10.1049/ip-f-2.1993.0015.
17. Doucet, A.; Godsill, S.; Andrieu, C. On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing* **2000**, *10*, 197–208. https://doi.org/10.1023/a:1008935410038.
18. Pitt, M.K.; Shephard, N. Filtering via Simulation: Auxiliary Particle Filters. *Journal of the American Statistical Association* **1999**, *94*, 590–599. https://doi.org/10.1080/01621459.1999.10474153.
19. Kantas, N.; Doucet, A.; Singh, S.S.; Maciejowski, J.; Chopin, N. On Particle Methods for Parameter Estimation in State-Space Models. *Statistical Science* **2015**, *30*. https://doi.org/10.1214/14-sts511.
20. Schon, T.; Gustafsson, F.; Nordlund, P.J. Marginalized particle filters for mixed linear/nonlinear state-space models. *IEEE Transactions on Signal Processing* **2005**, *53*, 2279–2289. https://doi.org/10.1109/tsp.2005.849151.
21. Storvik, G. Particle filters for state-space models with the presence of unknown static parameters. *IEEE Transactions on Signal Processing* **2002**, *50*, 281–289. https://doi.org/10.1109/78.978383.
22. Özkan, E.; Šmídl, V.; Saha, S.; Lundquist, C.; Gustafsson, F. Marginalized adaptive particle filtering for nonlinear models with unknown time-varying noise parameters. *Automatica* **2013**, *49*, 1566–1575. https://doi.org/10.1016/j.automatica.2013.02.046.
23. Lopes, H.F.; Tsay, R.S. Particle filters and Bayesian inference in financial econometrics. *Journal of Forecasting* **2010**, *30*, 168–209. https://doi.org/10.1002/for.1195.
24. CARTER, C.K.; KOHN, R. On Gibbs sampling for state space models. *Biometrika* **1994**, *81*, 541–553. https://doi.org/10.1093/biomet/81.3.541.
25. Andrieu, C.; Doucet, A.; Holenstein, R. Particle Markov Chain Monte Carlo Methods. *Journal of the Royal Statistical Society Series B: Statistical Methodology* **2010**, *72*, 269–342. https://doi.org/10.1111/j.1467-9868.2009.00736.x.
26. Del Moral, P., Feynman-Kac Formulae. In *Feynman-Kac Formulae*; Springer New York, 2004; pp. 47–93. https://doi.org/10.1007/978-1-4684-9393-1_2.
27. Urteaga, I.; Bugallo, M.F.; Djurić, P.M. Sequential Monte Carlo for inference of latent ARMA time-series with innovations correlated in time. *EURASIP Journal on Advances in Signal Processing* **2017**, *2017*. https://doi.org/10.1186/s13634-017-0518-4.

 doi:10.20944/preprints202507.2022.v1

28. Carlin, B.P.; Polson, N.G. Inference for nonconjugate Bayesian Models using the Gibbs sampler. *Canadian Journal of Statistics* **1991**, *19*, 399–405. https://doi.org/10.2307/3315430.
29. Avellaneda, M.; Lee, J.H. Statistical arbitrage in the US equities market. *Quantitative Finance* **2010**, *10*, 761–782. https://doi.org/10.1080/14697680903124632.
30. Khandani, A.E.; Lo, A.W. What Happened to the Quants in August 2007?: Evidence from Factors and Transactions Data. *SSRN Electronic Journal* **2008**. https://doi.org/10.2139/ssrn.1288988.
31. Jasra, A.; Singh, S.S.; Martin, J.S.; McCoy, E. Filtering via approximate Bayesian computation. *Statistics and Computing* **2010**, *22*, 1223–1237. https://doi.org/10.1007/s11222-010-9185-0.
32. Schoutens, W. *Lévy processes in finance*, reprinted ed.; Wiley series in probability and statistics, Wiley: Chichester [u.a.], 2005.
33. Murphy, K.P. *Machine Learning - A Probabilistic Perspective*; Adaptive Computation and Machine Learning, MIT Press: Cambridge, 2014. Description based upon print version of record.