Principal Component Analysis and Hidden Markov Model for Forecasting Stock Returns

Eugene W. Park

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science
Courant Institute of Mathematical Sciences
New York University
May, 2023

Abstract

This paper presents a method for predicting stock returns using principal component analysis (PCA) and the hidden Markov model (HMM), and tests the results of trading stocks based on this approach. Principal component analysis is applied to the covariance matrix of stock returns for companies listed in the S&P 500 index, and interpreting principal components as factor returns, we apply the HMM model on them. Then we use the transition probability matrix and state conditional means to forecast the factors returns. Reverting the factor returns forecasts to stock returns using eigenvectors, we obtain forecasts for the stock returns. We find that, with the right hyperparameters, our model yields a strategy that outperforms the buy-and-hold strategy in terms of the annualized Sharpe ratio.

Keywords: Principal component analysis, factor model, hidden Markov model, stock market, forecasting

1 Introduction

Stock market forecasting has been a prolonged practice of interest for people in various fields of discipline, and thus, various approaches have been used to tackle the problem from classical time series analysis and using factor models to high frequency trading and using deep learning techniques.

A widely used method to analyze time series data, the hidden Markov model has been a popular method to analyze financial markets. [1], [2], [4], and [5] have used the HMM on its own to make predictions of stock prices, while others have combined the HMM with other methods such as the long short term memory model [3] and fuzzy logic [6]. The usage of the hidden Markov model on stock price prediction is more extensive than what has been cited, but within the author's knowledge, there seems to be more attention on using HMM as a preliminary step than combining techniques that refine the input for the HMM.

In this paper, we present a method that applies principal component analysis (PCA) as a form of a factor model to preprocess our data and uses the HMM on the preprocessed data to forecast of stock returns. Then, we test the accuracy of this model by computing the directional accuracy of the forecasts, and evaluate the long-term, risk-adjusted returns of various trading strategies based on this model.

The rest of this paper is organized as follows. In section 2, we give a brief overview of the PCA as a factor model, and in section 3, briefly introduce the HMM. Then, we

explain our model in detail in section 4, and discuss the implementation and results of our model as trading strategies in section 5. We conclude the paper in section 6 by mentioning some shortcomings of our paper and providing ways for improvement.

2 Brief Introduction: Principal Component Analysis as Factor Models

2.1 Principal Component Analysis

Given a dataset $X \in \mathbb{R}^{t \times n}$ where the rows of X are de-meaned samples of the random vector $\tilde{X} \in \mathbb{R}^n$, principal component analysis is performed as follows:

- ◇ Compute the covariance matrix of X:

$$M = X^T X \in \mathbb{R}^{n \times n} \tag{1}$$

- ◇ Compute the eigendecomposition of M, which exists by the spectral theorem for symmetric matrices, such that

$$M = EGE^T$$

where $E \in \mathbb{R}^{n \times n}$ is the column matrix of n orthonormal eigenvectors $u_1, \ldots, u_n \in \mathbb{R}^n$, and $G \in \mathbb{R}^{n \times n}$ is a diagonal matrix of eigenvalues $\lambda_1, \ldots, \lambda_n$.

Since we are allowed to reorder the eigenvalues in descending order with the eigenvectors ordered accordingly, we may assume that

$$\lambda_1 > \cdots > \lambda_n$$

Each λ_i denotes the amount of variance explained in M, and u_i are the corresponding directions called the "principal directions."

Note that u_1 is the direction of the highest variance in M and u_n is the direction of the lowest variance.

The "principal components" are then the dataset X in the directions of the principal direction u_i:

$$w_i = \begin{bmatrix} w_{i,1} \\ \vdots \\ w_{i,n} \end{bmatrix} = Xu_i$$

2.2 Factor Models

The factor model, introduced in the Arbitrage Pricing Theory (APT) in [8], states that there exist explanatory variables (called "factors") that explain the systematic behavior of asset returns. Hence,

$$\mathbf{r} = B\mathbf{f} + \epsilon \tag{2}$$

where

- ◇ $\mathbf{r} \in \mathbb{R}^{n \times 1}$ is a random vector of returns for n assets,
- ◇ $\mathbf{f} \in \mathbb{R}^{k \times 1}$ is a random vector of returns for the k factors with $\mathbb{E}(\mathbf{f}) = \mu_f$ and $\sigma^2(\mathbf{f}) = F$,

- $\diamond$ $B \in \mathbb{R}^{n\times k}$ consists of columns representing factor loadings for the k factors,
- $\diamond$ $\epsilon \in \mathbb{R}^{n\times 1}$ is random noise for which we assume $\mathbb{E}(\epsilon) = 0$, $\sigma^2(\epsilon) = D$, a diagonal matrix.

From , 2 we get

$$\mathbb{E}(\mathbf{r}) = B\mu_{\mathbf{f}}$$
$$\sigma^2(\mathbf{r}) = \Sigma_{\mathbf{r}} = BFB^T + D \tag{3}$$

where $\Sigma_{\mathbf{r}} = \mathbb{E}(\mathbf{r}^T\mathbf{r}) \in \mathbb{R}^{n\times n}$ is the covariance matrix of $\mathbf{r}$.

2.3 Classical Factor Models and PCA Factor Models

[*] Suppose we have a dataset X of n asset returns across t time ($X \in \mathbb{R}^{t\times n}$). The classical, APT-type, factor models uses features of the assets to derive the k factors that explain X. Hence, factors are exogenous to X. The Fama-French five factor model [9] and factor-based risk models built by MSCI are some of the popular examples of the classic factor models.

The PCA factor models differ from the classical ones in that the factors are not discerned exogenously. The factors are data-driven (i.e. inferred from the data matrix X) by applying the eigendecompoistion on the covariance matrix in equation 1:

$$M = X^TX = EGE^T$$

The PCA factor model sets the eigenvectors as factor loadings,

$$E = B \tag{4}$$

, and the factor returns as the n principal components

$$\mathbf{f} = XE = [w_1, \dots w_n] \tag{5}$$

Then, we have

$$F = \mathbf{f}^T\mathbf{f} = (XE)^T(XE)$$
$$\Rightarrow EFE^T = E(XE)^T(XE)E^T = X^TX$$
$$\Rightarrow F = G$$

Assuming the PCA model to be exact, hence leaving aside D in (3) (i.e. $\epsilon = 0$ in (2)),

$$M = X^TX = EGE^T = BFB^T + 0 \tag{6}$$

Thus, the PCA factor model defines n factors that explain the full covariance matrix of our dataset of returns, X, where each factor corresponds to X rotated in n independent direction $u_i \in E$, $i = 1, \dots, n$, and the factor loadings are these n independent directions. [†] The analogue to (2) is then

$$X^T = E(XE)^T \tag{7}$$

where $X^T \in \mathbb{R}^{n\times t}$ and $(XE)^T \in \mathbb{R}^{n\times t}$ consists of t samples in the columns.

[*] This subsection is largely based on [7]

[†] Note that the independence of eigenvectors come from the spectral theorem of symmetric matrices, and that the independence of u_i's suggests that $f|_k$ consists of independent column vectors

We note that the although factors derived by PCA are less intuitive, the PCA offers the benefit of discovering factors that not may be discovered exogenously.

3 Brief Introduction: Hidden Markov Model

The fundamental argument of the hidden Markov model (HMM) is that underlying the observed sequence of time series data $Y = \{y_1, \ldots, y_t\}$, there exists a Markov chain, $Z = \{z_1, \ldots, z_t\}$ that generates Y where each $z_i \in S = \{1, ..., N\}$, the state space. The underlying Markov chain has an initial distribution $\mathbf{L} = \{L_1, \ldots, L_n\}$ and a transition probability matrix $\mathbf{P} = (p_{ij})_{i,j \in S}$, where p_{ij} denotes the probability of z_l transitioning from state i to j. Finally, assuming our observations to be in state space O, the emission probability matrix, $\mathbf{R} = (r_{ij})_{i \in S, j \in O}$, denotes the probability of observation given that we are in a certain state: $r_{ij} = \mathbb{P}(Y = j | Z = i)$.

Hence, the parameters that define the HMM are:

$$\Theta = (\mathbf{L}, \mathbf{P}, \mathbf{R}) \tag{8}$$

and we are mainly concerned with the following problems:

1. Selecting the best model Θ given a range of model options, Θ_k and the sequence of observations Y:
$$\underset{\Theta_k}{\text{argmax}} \mathbb{P}(Y | \Theta_k)$$

2. Determining the most probable state sequence Z given Y and Θ:
$$\underset{Z}{\text{argmax}} \mathbb{P}(Z | Y, \Theta)$$

3. Estimating the parameters Θ given Y:
$$\underset{\Theta}{\text{argmax}} \mathbb{P}(Y | \Theta)$$

The *forward algorithm* is an algorithm to solve the first problem, *Viterbi algorithm* for the second, and *Baum-Welch Algorithm* or EM (expectation-maximization) algorithm for the last.[‡]

4 PCA + HMM as a Forecasting Model

In this section, we discuss details for how we use PCA with HMM to develop a model that forecasts asset returns.

Suppose that we are working with a dataset, X, of returns for n assets through T time periods such that the columns of X are time series data of return for a particular asset:

$$X = (r_{ij})_{i=\{1,\ldots,T\}, j=\{1,\ldots,n\}}$$

, where r_{ij} is the return of company j at time i. Thus, our goal is to forecast the returns on the n assets for the next period, $\hat{X}_{T+1, n=\{1,\ldots,n\}}$. Furthermore, since the directional accuracy of our forecasts is crucial for our trading strategies, we are, in fact, primarily concerned with:

$$\text{sign}(\hat{X}_{T+1, n=\{1,\ldots,n\}}) \tag{9}$$

[‡]Refer to [10] for details of the algorithms

4.1 Implementing PCA

PCA only requires our dataset to be de-meaned, but we normalize X for each column and denote it as Y:

$$Y = \frac{X - \mu_x}{\sigma_X} \tag{10}$$

We apply PCA by computing the eigendecomposition of the covariance matrix of Y such that the eigenvalues are in decreasing order and the eigenvectors are ordered accordingly:

$$H = Y^T Y = EGE^T, \ H, E, G \in \mathbb{R}^{n \times n}$$
$$\text{diag}G = \{\lambda_1, \ldots, \lambda_n\}, \ \lambda_1 > \cdots > \lambda_n$$

Then, as in (7), we have a full PCA factor model:

$$Y^T = E(YE)^T$$

We assume that the covariance matrix H, or equivalently Y, contains some noise. Thus, we want to de-noise our dataset before training the HMM.

Recall that the eigenvalues suggest the percentage of variance (i.e. information) of H explained by rotating it in the direction of the corresponding eigenvectors. Suppose the amount of noise in the covariance matrix is $p\%$. Then, we take the first k eigenvectors that explain at least $(1-p)\%$ of variance such that

$$\lambda_1 + \cdots + \lambda_k \geq 1 - p$$

$$\lambda_1 + \cdots + \lambda_{k-1} < 1 - p$$

and compute the corresponding principal components, $(w_1, \ldots, w_k)$, which represent k of the n factor returns as in (5).

Then, we extract $\approx p\%$ [§] of noise from H by restricting our set of eigenvectors, E, to the first k eigenvectors

$$E|_k = \{u_1, \ldots, u_k\} \in \mathbb{R}^{n \times k}$$

and computing k factor returns,

$$f|_k = \{w_1, \ldots, w_k\} = YE|_k = \begin{bmatrix} y_1 \cdot u_1 & \ldots & y_1 \cdot u_k \\ y_2 \cdot u_1 & \ldots & y_2 \cdot u_k \\ \vdots & \vdots & \vdots \\ y_T \cdot u_1 & \ldots & y_T \cdot u_k \end{bmatrix} \in \mathbb{R}^{T \times k} \tag{11}$$

, where $y_i \in \mathbb{R}^k$ is the returns for the k factors in time i. Note that $f|_k$ consists of columns that represent Y in k independent directions with the most amount of information (i.e. $f|_k$ consists of independent column vectors).

Assuming the full PCA factor model to contain some noise, we are essentially assuming that not all of the principal component factors defined by PCA are significant factors; factors that explain small portions of variance of the covaraince matrix are just noise. Hence, we have:

$$Y^T = E(YE)^T = E|_k(YE|_k)^T + \epsilon_k \tag{12}$$

[§] $\approx p\%$ is due to the fact that the k eigenvectors does not necessarily, in fact, most often does not, explain exactly $(1-p\%)$ of the matrix.

where ϵ_k is the error term that contains $\approx p\%$ of information in H.

4.2 HMM on Factor Returns

Now, we extract the noise from Y and apply the HMM to $f|_k$, the time series of k factor returns, to obtain a one-step-ahead forecast for these factor returns. Then, we revert the forecast of the factors back to assets which will then be our forecast for the assets.

Before, we begin training the model, we assume the emission probability in each state in the state space, S, to have have a Gaussian distribution. Hence, the parameters of our model are

$$\Theta = (\mathbf{L}, \mathbf{P}, \mathbf{R})$$

$$\Rightarrow \Theta = (\mathbf{L}, \mathbf{P}, \mu_i, \sigma_i^2), i = 1, \ldots, N \tag{13}$$

, where N is the number of states in the state space S, μ_i is the mean of the observation sequence in state i, and σ_i^2 is the variance of observation sequence in state i.

4.2.1 Training & Model Selection

To determine the number of state space, N, we train the Gaussian HMM (13) for each j in [2,3,4,5,6,7,8], denoted as Θ_j, compute the log likelihood, $\ln(\mathbb{P}(Y|\Theta_j))$, using the forward algorithm, and compute Akaike information criterion (AIC)[11] for each model.

$$\text{AIC} = -2\ln(\ln(\mathbb{P}(Y|\Theta_j))) + 2j$$

Then, we choose our model to be Θ_j, which yields the lowest AIC.

In the training step of each Θ_j, we use the Baum-Welch algorithm to estimate the parameters, and thus, choose the initial conditions for our parameters as follows:

$$\mathbf{L}_0 = 1/j \tag{14}$$

$$\mathbf{P}_0 = 1/j^2 \tag{15}$$

$$\mu_{i,0} = \mathbb{E}(o)_i \tag{16}$$

$$\sigma_{i,0}^2 = \text{var}(o)_i \tag{17}$$

, where o is the observation sequence, $o = \{o_1, \ldots, o_T\}$. Here, we are assuming that it is equally likely for the underlying Markov chain to start in any state (14) and that it is equally likely for the Markov chain to transit from one state to another (15). (16) suggests that we assume the μ_i to be the sample mean of our observations for all states $i = 1, \ldots, j$ (i.e. same initial mean for all states), and (17) suggests that we assume σ_i^2 to be the sample variance of our observations for all states $i = 1, \ldots, j$ (i.e. same initial variance for all states).

Note that since $f|_k$ consists of k time series of factor returns, our observation sequnce is k-dimensional.

$$o = \{o_1, \ldots, o_T\}, \text{where } o_t = \{o_t^1, \ldots, o_t^k\}$$

Hence, σ_i^2 is a $k \times k$ covariance matrix. Since, the k factor returns are independent, however, we may assume the covariance matrix to be diagonal. Thus, in (17), $\sigma_{i,0}^2 = \text{var}(o)_i$ is a $k \times k$ diagonal covariance matrix, for state i, with sample variances of $o = \{o^1, \ldots, o^k\}$ in the diagonal.

4.2.2 Forecasting

Now, once we have found and trained our model, which we will denote again as Θ, we use the estimated parameters to forecast the returns of the k factors in the next time period, $(\hat{f}|_k)_{T+1}$, as follows:

1. We use the *Viterbi algorithm* to obtain the sequence of the underlying Markov chain:
$$Z = \{z_1, \ldots, z_T\},\ z_i \in S = \{1, \ldots, N\}$$
, where $z_T = i$ states that the underlying Markov chain is currently in state i.

2. Then, we use the transition probability matrix, $\hat{\mathbf{P}}$, and the state conditional mean, $\hat{\mu}_i$, to obtain an estimate for $(f|_k)_{T+1}$:
$$(\hat{f}|_k)_{T+1} = \sum_{j \in S}^{N} P_{ij}\hat{\mu}_j \tag{18}$$
Since $\hat{\mu}_j \in \mathbb{R}^k$, for all $j = \{1, \ldots, N\}$, $(\hat{f}|_k)_{T+1} \in \mathbb{R}^k$.

Thus, our forecast of the k factor returns in the next period depends on 1) how likely it is to transition from state i to j, and 2) estimated state conditional means of the k factors.

Although we may trade factors, especially if it were defined exogenously as in the classical factor models, it is hard to trade endogenous factors defined by the PCA. Hence, we revert the forecasts for the factor returns to forecasts for the n assets. I.e. we want:
$$(\hat{f}|_k)_{T+1} \to \hat{Y}_{T+1}$$
Recall from (12) that
$$Y^T = E(YE)^T = E(f)^T$$
where $f = \{w_1, \ldots, w_n\}$. Since we only have an estimate of $f|_k$ for time $T+1$, we first estimate $f_{k+1}, \ldots, f_n$ for $T+1$. Recall that the best estimate is its mean. Thus, we have, for $j = k+1, \ldots, n$,
$$(\hat{f}_j)_{T+1} = \mathbb{E}((f_j)_{T+1}) = \frac{1}{T}\sum_{i=0}^{T}(Y_i \cdot u_j) = \frac{1}{T}\sum_{i=0}^{T}(Y_i) \cdot u_j = 0 \tag{19}$$
where $Y_i \in \mathbb{R}^n$ is the i-th row of Y, and the last equality follows from Y being normalized with row mean equal to 0. Equivalently,
$$\mathbb{E}((\hat{\epsilon_k})_{T+1}) = \mathbb{E}(\hat{Y}^T_{T+1} - E|_k(\hat{Y}_{T+1}E|_k)^T) = \frac{1}{T}\sum_{h=0}^{T} Y_h^T - E|_k(\frac{1}{T}\sum_{h=0}^{T} Y_h^T)E|_k)^T = 0$$
where Y_h^T are the h-th column of Y^T.

Now, we have
$$\hat{f}_{T+1} = \{(\hat{f}|_k)_{T+1}, 0, \ldots, 0\}$$
and from (12), we have
$$\hat{Y}^T_{T+1} = E(\hat{f}_{T+1})^T = E|_k(\hat{f}|_k)^T_{T+1}$$

$$\Rightarrow \hat{Y}_{T+1} = (\hat{f}|_k)_{T+1} E|_k^T$$

where $\hat{Y}_{T+1} \in \mathbb{R}^{1\times n}$, $(\hat{f}|_k)_{T+1} \in \mathbb{R}^{1\times k}$, and $E|_k^T \in \mathbb{R}^{k\times n}$. Thus, we obtain forecasts for the n assets by multiplying the transpose of the k eigenvectors to the forecasts for the k factor returns.

Finally, since Y is the normalized version of X (10),

$$\hat{X}_{T+1} = \sigma_X(\hat{y}_{T+1}) + \mu_X \tag{20}$$

5 Implementation

We use data for the weekly returns of companies listed in the S&P500 to train our model.[¶][‖] The returns are calculated using the closing stock prices, and each return sequence comprise the columns of our dataset, denoted X. Hence, we have $X \in \mathbb{R}^{Txn}$, where T is the last period of the time span of our data and n is the number of assets.

We note that, as companies are newly listed and delisted from the S&P500 index, some companies that are listed in the index at the time of data retrieval may not have been listed previously, and some may have been delisted during the time span of our data. Thus, we are subject to *survivor bias*, but minimize this effect by ensuring that we have at least 400 data points of company returns for each time period. With at least 80% of the index, we conjecture that the several companies we fail to capture only explains a minimal amount of information/variance in the covariance matrix, X^TX, which we implement the PCA on. In other words, we suspect that most of the information from the companies we fail to capture have already been captured by the eigenvectors corresponding to large eigenvalues, and that the additional information from those companies constructs eigenvectors with very small eigenvalues. Thus, we suspect that they are eventually extracted during the process of denoising our covariance matrix via PCA. Because this is only a conjecture, however, we do not know for sure how much we are affected by survivor bias, and thus, leaving room for improvement upon this paper.

5.1 Model Training

Ensuring our dataset to include returns data for at least 400 companies listed in the S&P500 index for each t, we start our data from July 23, 2004. Using 10 years' worth of weekly data, we train our PCA+HMM model and forecast the weekly returns for the next period. Then, rolling the 10 years window by one week[**], we forecast returns for a total of 100 weeks (the week for which we make our last forecast is July 8, 2016, thus our data spans from July 23, 2004 to July 8,2016)[††]. Note that, since we train the PCA+HMM for each time window, the total number of PCA factors, the total number of states in the HMM, and the model parameters may vary over time.

Since PCA is essentially an eigendecomposition, We use the *linalg* module in the *numpy* library in Python to implement PCA. To train the HMM model, we use the *gaussianhmm* module in the *hmmlearn* Python library.

Recall from part 4.1 that we have to specify the hyperparameter, p, that specifies the minimum percentage of noise to be extracted during the dimension reduction process.

[¶] Data was obtained from finance.yahoo.com

[‖] List obtained from https://en.wikipedia.org/wiki/List_of_S%26P_500_companies on April 28, 2023

[**] The exact time samples (rows of our dataset) for each window may slightly differ due to different numbers of holidays and trading days every year. Moreover, due to the entering and exiting of companies from the index since the beginning of our dataset, the number of stocks (columns of our dataset) may also differ.

[††] Additional data is available, but we limit ourselves to 100 forecasts for computational convenience.

We train our model with four choices of p:

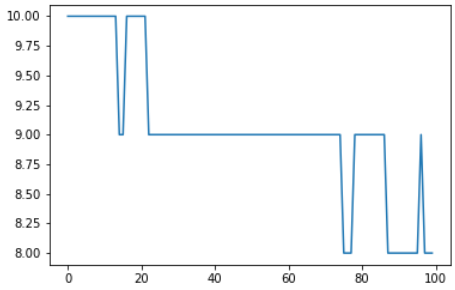
$$p = \{45\%, 30\%, 15\%, 10\%\}$$

which corresponds to keeping no less than 55%, 70%, 85%, and 90% of the information in Y, our normalzied return matrix.

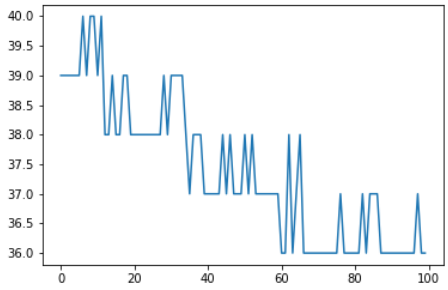
5.1.1 Model Implications

Before, we test trading strategies based on our model, we make several observations of our trained model.

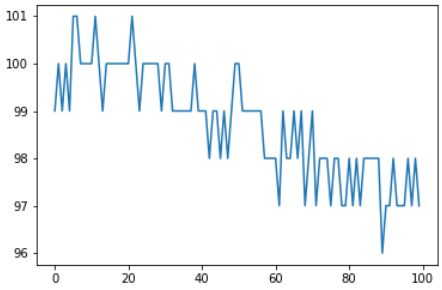
Figure 1 shows the total number of PCA factors kept in the model throughout time for the different choices of p. Notice that as we keep more noise in our model, the number of PCA factors kept not only increases, but also changes more frequently. Moreover, there is also a more gradual change in the total number of PCA factors in the market, such as from the first 20 periods to the period between the 20th and 60th week in figures 1.a and 1.b. There is also a gradual shift in 1.c and 1.d from the first 50 weeks to the next. This suggests that there may exist short-term PCA factors and longer-term PCA factors.



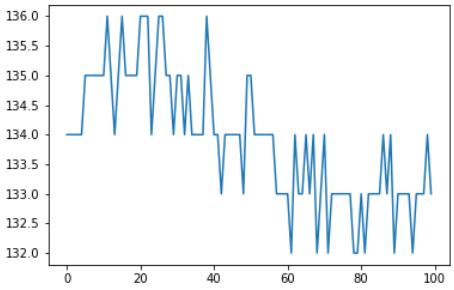
(a) P = 45%



(b) P = 30%



(c) P = 15%



(d) P = 10%

Figure 1: Number of PCA Factors

Now, we observe the state-conditional means and variances calibrated from the HMM model. Recall that the HMM calibrates the means and variances for K PCA factors for each state, and since we run the model for 100 periods, we have a series of state-conditional means and variances as follows:

$$\mu = \{\mu_1, \ldots, \mu_{100}\}$$

$$\sigma^2 = \{\sigma_1^2, \ldots, \sigma_{100}^2\}$$

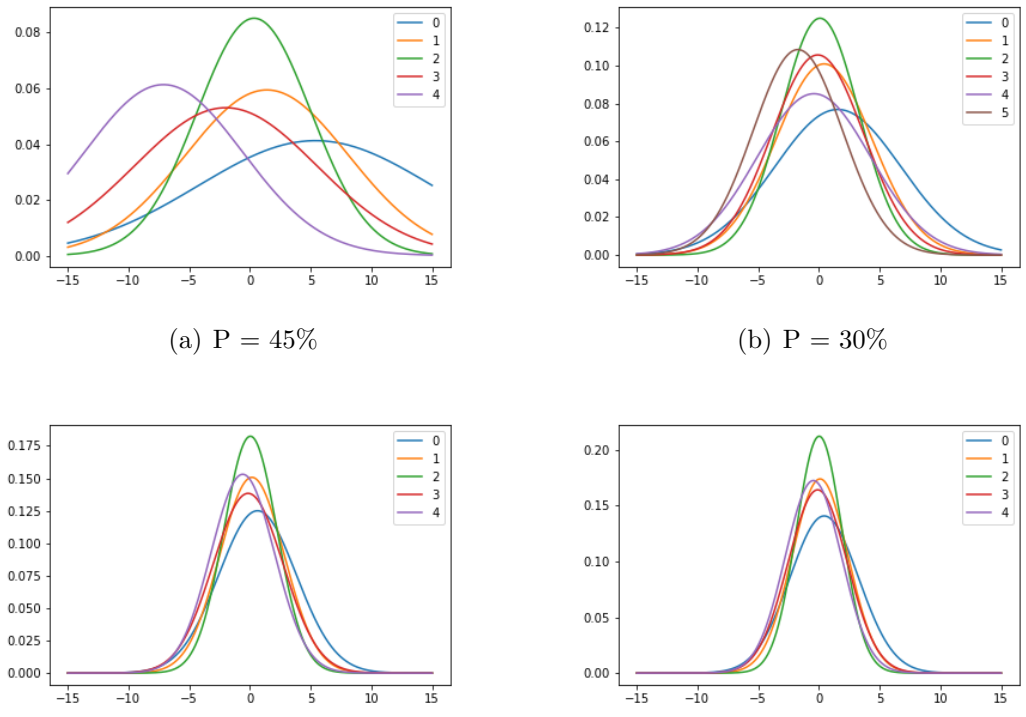
where $\mu_t, \sigma_t^2 \in \mathbb{R}^{N_t \times k_t}$. Note that N_t and k_t has a time subscript since the total number of states and PCA factors vary.

At each time, t, we average μ_t, σ_t^2 along the k factors to get a time series of state-conditional means and variances (of returns)

$$\tilde{\mu} = \{\tilde{\mu}_1, \ldots, \tilde{\mu}_{100}\}$$

$$\tilde{\sigma}^2 = \{\tilde{\sigma}_1^2, \ldots, \tilde{\sigma}_{100}^2\}$$

where $\tilde{\mu}_t, \tilde{\sigma}_t^2 \in \mathbb{R}^{N_t}$. Moreover, since the total number of states are changing, we sort $\tilde{\mu}_t$ in decreasing order with $\tilde{\sigma}_t^2$ having the corresponding order. Thus, we disregard the actual state numbers, such as rather we are in state 1 or state 2, and put meaning on the states by ranking them according to the calibrated state-conditional means averaged across the k_t PCA factors. For each p, we find the minimum number states that existence throughout time and visualize the emission probability distributions in figure 2.



(a) P = 45%

(b) P = 30%

(c) P = 15%

(d) P = 10%

Figure 2: State-Conditional Emission Probability Distributions
Note:x-axis in percentages. 3 refers to 300%.

From figure 2, we can see that more noise being kept in our factor returns yields probability distributions that are less spread across states, suggesting that the underlying states lose significance and the HMM model becomes less useful. Furthermore, figure 2.a, the model with 45% of noise extracted from the normalized return dataset,Y, is intuitively interpretable. State 4 to seems to be "bear market" state, while state 0 to can be viewed as the "bull market" state, and states 1,2, and 3 are the relatively

stable states of the market. We can also view state 4 as where investors have excessively fearful, state 0 as the state where they are excessively optimistic, states 1 and 3 as states where they are moderately optimistic and fearful, and state 2 as where they neutral about the market. Thus, models with more spread-out probability distributions provide more reliable and interpretable information about the market which may be useful for task such as risk-management.

5.2 Trading Strategies

Now we construct and test simple trading strategies based on our model.

1. Our first trading strategy is solely based on the forecasts given by **??**. We short the assets for which $\hat{X}_{T+1}$ predicts a negative return and long the ones for which $\hat{X}_{T+1}$ predicts a positive return. We weight the n stocks equally (i.e. 1/n) and trade at the closing time of the stock market at the last trading day of every week.

2. The second strategy depends on the forecasts of the normalized asset returns, $\hat{y}_{T+1}$. Since
$$Y = \frac{X - \mu_X}{\sigma_X}$$
, we hypothesize that there is predictive trading signal in not the return itself, but the excess return over the average.

When testing our trading strategies, we assume the following:

1. we are able to trade at exactly the closing time
2. we can exactly match the n assets with equal weight in our portfolio
3. and we do not incur any trading costs.

5.3 Results

To evaluate the performance of our trading strategies, we compare our trading strategies with the simple buy-and-hold strategy of all companies in the index. As measurements of performance, we compute:

1. the "winning probability" of our forecasts (i.e. probability of correctly forecasting the direction of the returns in the next period).
2. the annualized Sharpe ratio:
$$S = \frac{r - rf}{\sigma} \tag{21}$$
where r is the asset return, rf is the return of the risk-free rate, and σ is the standard deviation of asset. We will assume the risk-free rate to be zero.

The winning probabilities of the two trading strategies are shown in table 1. We observe that strategy 1 clearly outperforms strategy 2 in forecasting the direction of the returns. Yet, strategy 1 yield just above 50%, suggesting that significant profit can be made only in a casino-style type of trading where the model trades constantly without going bankrupt. Hence, for this strategy to be profitable over the long-run, additional features must be included such as the trade size.

Table 2 shows the Sharpe ratios of the trading strategies, and we observe that the strategy with the highest Sharpe ratio is strategy 2 with $p = 15\%$. This outperforms

Table 1: Winning Probability

p%	Strategy 1	Strategy 2
45%	0.532	0.490
30%	0.543	0.504
15%	0.538	0.511
5%	0.532	0.490

Note: Rounded to the nearest thousandth.

the buy-&-hold strategy by more than 50%. From figure 5, we observe that the strategy yields lower cumulative return than the buy-&-hold, but is more stable; it rides out two ocassions of downturns, during late-2014 and early-2016, without much losses, contrary to the other strategies. It also has a winning probability of 51%, and thus, seems to be useful in terms of risk management and as a stable component within a portfolio.

We also note that the Sharpre ratios depend heavily on the hyperparameter p. The difference between the highest and lowest ratios of strategy is as high as 0.91. For strategy 1, there seems to be a trend where the ratio falls as p decreases to 30%, but increases again as more noise is extracted. This suggests the significance of finding the right hyperparameter p, which could yield significantly higher Sharpe ratios than the ones presented here.

Table 2: Annualized Sharpe Ratio

p%	Strategy 1	Strategy 2
45%	0.688	0.45
30%	0.581	0.581
15%	0.703	1.36
10%	0.877	0.726
Buy-&-Hold	0.828	

Note: Values are percentages.

6 Conclusion

We have shown that with the right hyperparameter p, the PCA + HMM model provides an intuitive understanding of the market through the estimated emission probability distributions of the states, and yields model-based trading strategies with forecast accuracy slightly above 50%. We have also seen that these strategies have the potential to outperform the market by finding the right p. This work suggests that future research into finding the optimal p will be very useful. Moreover, we have made several assumptions which such as zero transaction costs and that we are able to trade at exactly the closing prices at the closing time. These assumptions are unrealistic and fails in practice. Hence, our paper could be improved by evaluating the performance under realistic conditions, such as existence of slippage, transactions costs, and possibly price impact as well (when we assume large trade size).

Acknowledgement: We thank professors Kenneth Winston, Petter Kolm, and Jonathan Goodman for their valuable comments and suggestions throughout the progress of this work.

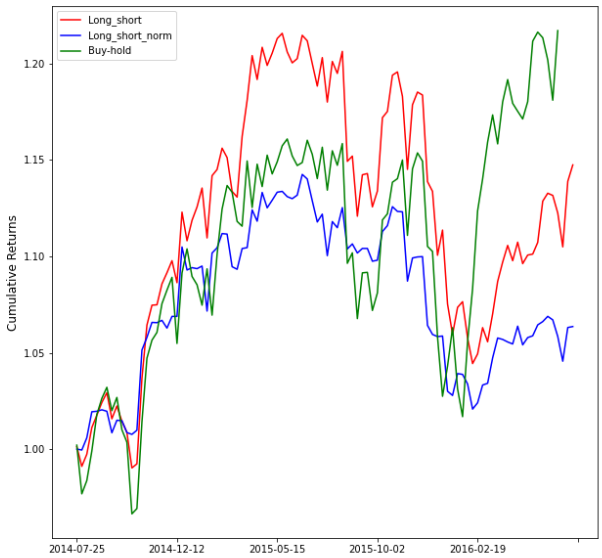


Figure 3: Return Sequences for Strategies 1 and 2 with p = 45%

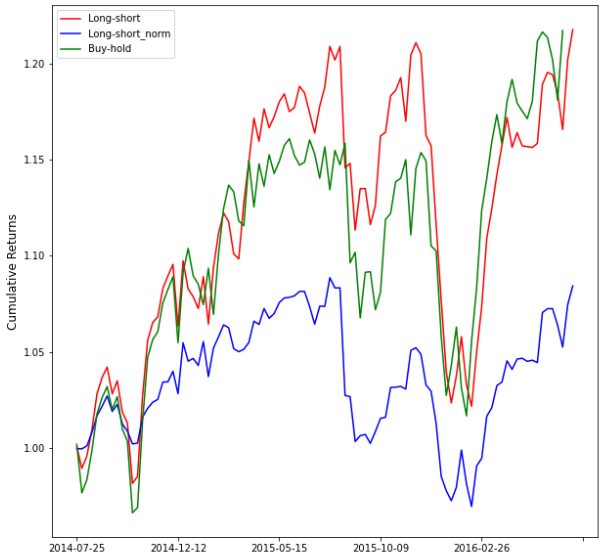


Figure 4: Return Sequences for Strategies 1 and 2 with p = 30%

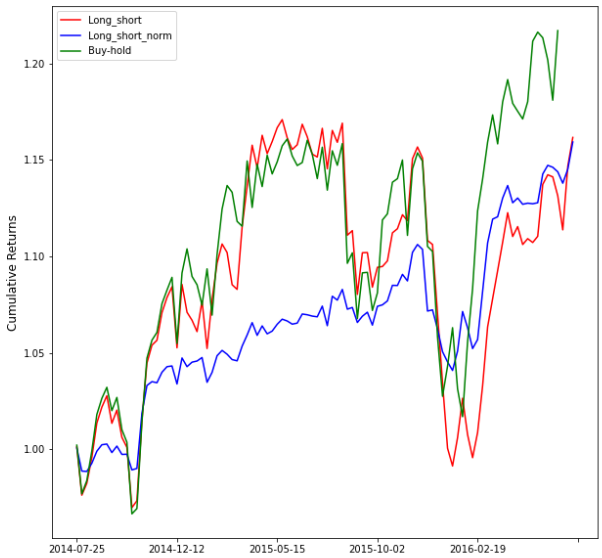


Figure 5: Return Sequences for Strategies 1 and 2 with p = 15%

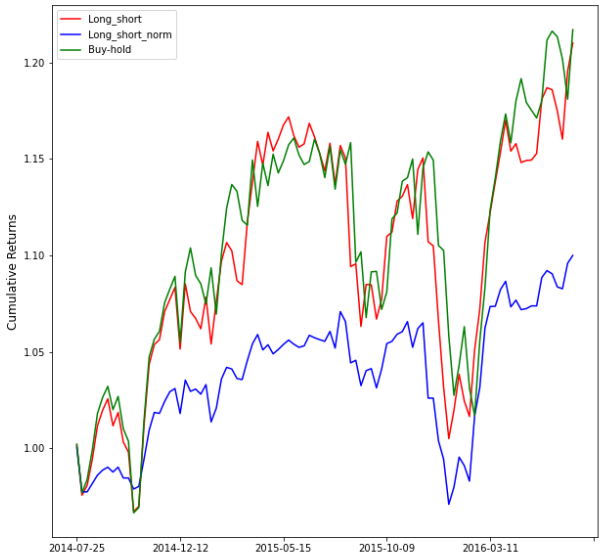


Figure 6: Return Sequences for Strategies 1 and 2 with p = 10%

Appendix

Appendix A.1. Proof of Forecast Equation 18:

Denote r_t, return at time t, as the rows of f_t.

$$\begin{aligned}
(\hat{f}|_k)_{T+1} &= \mathbb{E}(r_{T+1}|Z_T = i) \\
&= \sum_{j \in S}^{N} \mathbb{E}(r_{T+1}, Z_{T+1} = j|Z_T = i) \\
&= \sum_{j \in S}^{N} \mathbb{E}(r_{T+1}|Z_{T+1} = j)\mathbb{P}(Z_{T+1} = j|Z_T = i) \\
&= \sum_{j \in S}^{N} P_{ij}\hat{\mu}_j
\end{aligned}$$

References

[1] N. Nyugen. "Hidden Markov Model for Stock Trading". *International Journal of Financial Studies,* pages 6-36, 2018.

[2] B. Dhingra, A. Gupta. "Stock Market Prediction Using Hidden Markov Models". *2012 Students Conference on Engineering and Systems,* pages 1-4, 2012.

[3] J. Huo, M. Liu, Y. Wu, J. Wu. "Stock Market Trend Analysis Using Hidden Markov Model and Long Short Term Memory". *eprint arXiv:2104.09700*, 2012.

[4] R. Hassan, B. Nath. "Stock market forecasting using hidden Markov model: a new approach". *5th International Conference on Intelligent Systems Design and Applications,* pages 192-196, 2005.

[5] G. Kavitha, A. Udhayakumar, D. Nagarajan. "Stock Market Trend Analysis Using Hidden Markov Models". *International Journal of Computer Science and Information Security ,* 2013.

[6] R. Hassan. "A Combination of hidden Markov model and fuzzy model for stock market forecasting". *Neurocomputing,* pages 3439-3446, 2009.

[7] K. J. Winston. Quantitative Risk and Portfolio Management: Theory and Practice. Cambridge University Press, 2023 (forthcoming).

[8] S. A. Ross. "The arbitrage theory of capital asset pricing". *Journal of Economic Theory,* pages 341-360, 1976.

[9] E. F. Fama, K.R. French. "A five-factor asset pricing model". *Journal of Financial Economics,* pages 1-22, 2014.

[10] T. Li, W. E, E. Vanden-Eijnde. Applied Stochastic Analysis. American Mathematical Society, 2019.

[11] A. Hirotugu. "A new look at the statistical model identification". *IEEE TRansactions on Automatic Control,* pages 716-723, 1974.