Prediction of stock return series with hidden Markov models

Tian Gao

June 9, 2016

Supervised by Associate Professor Jianzhong Lin
School of Mathematical Sciences
Shanghai Jiao Tong University

Table of Contents

1. Introduction

2. Preliminary Knowledge

3. Hidden Markov Model

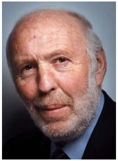
4. Stock Return Series Prediction System

5. Empirical Analysis

6. Summary

Introduction

Background



- Quantitative finance analysis develops in China.
- Simple time series analysis techniques are incapable of capturing the market states.
- Dr. James H. Simons implemented hidden Markov models (HMM) at Medallion Fund of Renaissance Technologies LLC.

Major Contributions and Key Spotlights

Major Contributions

- Construct the entire system for stock return series prediction.
- Carry out empirical analysis on both Chinese and U.S. stock markets, and among data with different observation frequencies.

Key Spotlights

- The system thoroughly encapsulates the modules of data-preprocessing, model initialization, parameters estimation and model calibration, hidden states decoding and analysis, return series prediction and results output.
- Hidden states analyses match common acknowledgements, and the system performs well in adaptive stock return series predictions.

Thesis Organization

The thesis includes 7 chapters and 2 appendices:

- Chapter 1. Introduction
- Chapter 2. Preliminary Knowledge and Models
- Chapter 3. Hidden Markov Models
- Chapter 4. Stock Return Series Prediction System
- Chapter 5. Empirical Analysis on Stock Market Indices
- Chapter 6. Conclusion
- Chapter 7. Future Work
- Appendix A. Visualization of Simulated Prediction Results
- Appendix B. Model Realization Python Codes

Preliminary Knowledge

Independent Mixture Distributions

Independent mixture distribution can deal with multi-modality of data. Well-known mixture models include exponential distribuiton family models [Hasselblad(1969)], of which the most famous is Gaussian mixture model (GMM) [Behboodian(1970)].

An independent mixture distribution is formulated as:

$$p(x) = \sum_{i=1}^{m} \delta_i p_i(x), \quad \text{s.t. } \sum_{i=1}^{m} \delta_i = 1,$$

where p_i are component distributions and δ_i are component weights.

Markov Chain

- Markovian property:

$$\begin{aligned}&\mathbb{P}(X_{t_n} \leq x_n \mid X_{t_1} = x_1, X_{t_2} = x_2, \ldots, X_{t_{n-1}} = x_{n-1})\\ =&\mathbb{P}(X_{t_n} \leq x_n \mid X_{t_{n-1}} = x_{n-1}).\end{aligned}$$

- Homogeneity:

$$\mathbb{P}(X_{m+k} = j \mid X_m = i) = \mathbb{P}(X_k = j \mid X_0 = i) := \gamma_{ij}^{(k)}.$$

- k-step transition probability matrix:

$$\mathbf{\Gamma}^{(k)} = \left(\gamma_{ij}^{(k)}\right)_{i,j \in \mathbb{S}}.$$

- Chapman-Kolmogorov equation:

$$\mathbf{\Gamma}^{(t+u)} = \mathbf{\Gamma}^{(t)}\mathbf{\Gamma}^{(u)}.$$

K-Means Clustering

Proposed by Stuart Lloyd in 1957 and by E. W. Forgy in 1965 [Forgy(1965)] separately, known as Lloyd-Forgy algorithm. The idea originates from vector quantization (VQ) in signal processing theories.

- Problem formulation:

$$\min_{\mathbf{S}} \sum_{i=1}^{k} \sum_{\mathbf{x} \in S_i} \|\mathbf{x} - \boldsymbol{\mu}_i\|^2,$$

- Algorithm:
  1. assign step:

$$S_i^{(t)} = \left\{ x_p : \|x_p - m_i^{(t)}\|^2 \le \|x_p - m_j^{(t)}\|^2, \forall j,\ 1 \le j \le k \right\}.$$

  2. update step:

$$m_i^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum_{x_j \in S_i^{(t)}} x_j.$$

Hidden Markov Model

Model Formulation

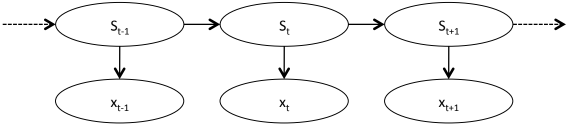


Figure 1: Directed graph of a typical HMM

- A HMM is a doubly stochastic process, explained with details in [Zucchini and MacDonald(2009)].
- The hidden states process $\{S_t \colon t = 1, 2, \dots\}$ is a Markov chain:

$$\mathbb{P}(S_t \mid \mathbf{S}^{(t-1)}) = \mathbb{P}(S_t \mid S_{t-1}),\ t = 2, 3, \dots .$$

- The observed variable are conditionally distributed:

$$\mathbb{P}(X_t \mid \mathbf{X}^{(t-1)}, \mathbf{S}^{(t-1)}) = \mathbb{P}(X_t \mid S_t),\ t = 1, 2, 3, \dots .$$

Primary Statistics

- State-dependent distribution (conditional distribution of X):

$$p_i(x) = \begin{cases} \mathbb{P}(X_t = x \mid S_t = i) & \text{, if } X_t \text{ is discrete,} \\ f_t(x \mid S_t = i) & \text{, if } X_t \text{ is continuous.} \end{cases}$$

- Marginal distribution:

$$\mathbb{P}(X_t = x) = \sum_{i=1}^{N} \mathbb{P}(X_t = x \mid S_t = i)\mathbb{P}(S_t = i) = \sum_{i=1}^{N} p_i(x)\delta_i(t),$$

or in form of matrix:

$$\mathbb{P}(X_t = x) = (\delta_1(t), \delta_2(t), \dots, \delta_N(t)) \begin{pmatrix} p_1(x) & & 0 \\ & \ddots & \\ 0 & & p_N(x) \end{pmatrix} \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}$$

$$= \boldsymbol{\delta}(t)\mathbf{P}(x)\mathbf{1}' = \boldsymbol{\pi}\boldsymbol{\Gamma}^{t-1}\mathbf{P}(x)\mathbf{1}',$$

Primary Statistics

- Likelihood:

$$L_T = \boldsymbol{\pi}\mathbf{P}(x_1)\boldsymbol{\Gamma}\mathbf{P}(x_2)\boldsymbol{\Gamma}\mathbf{P}(x_3)\cdots\boldsymbol{\Gamma}\mathbf{P}(x_T)\mathbf{1}'.$$

- Forward probabilities:

$$\boldsymbol{\alpha}_t = \boldsymbol{\pi}\mathbf{P}(x_1)\boldsymbol{\Gamma}\mathbf{P}(x_2)\boldsymbol{\Gamma}\mathbf{P}(x_3)\cdots\boldsymbol{\Gamma}\mathbf{P}(x_t) = \boldsymbol{\pi}\mathbf{P}(x_1)\prod_{s=2}^{t}\boldsymbol{\Gamma}\mathbf{P}(x_s),$$

$$\alpha_t(j) = \mathbb{P}(\mathbf{X}^{(t)} = \mathbf{x}^{(t)}, S_t = j).$$

- Backward probabilities:

$$\boldsymbol{\beta}_t = \boldsymbol{\Gamma}\mathbf{P}(x_{t+1})\boldsymbol{\Gamma}\mathbf{P}(x_{t+2})\cdots\boldsymbol{\Gamma}\mathbf{P}(x_T)\mathbf{1}' = \left(\prod_{s=t+1}^{T}\boldsymbol{\Gamma}\mathbf{P}(x_s)\right)\mathbf{1}',$$

$$\beta_t(i) = \mathbb{P}(\mathbf{X}^{(t+1:T)} = \mathbf{x}^{(t+1:T)} \mid S_t = i),\ \mathbb{P}(S_t = i) > 0$$

Expectation Maximization Algorithm

- Specifically known as Baum-Welch algorithm in the context of HMM, proposed separately in [Baum and Petrie(1966)],[Baum and Eagon(1967)] and [Baum et al.(1970)].
- With forward and backward procedure, the algorithm has two steps:
  1. **E step** calculates the expectations of (functions of) the missing data conditional on the observations and the current estimate of $\boldsymbol{\theta}$;
  2. **M step** maximizes the CDLL w.r.t. $\boldsymbol{\theta}$.
- Essentially EM estimations are also a kind of MAP estimation, i.e. maximize a posterior estimations.

Expectation Maximization Algorithm

- State value indicators:

$$u_j(t) = \begin{cases} 1 & \text{, iff } S_t = j \\ 0 & \text{, otherwise,} \end{cases} \quad \text{and} \quad v_{jk}(t) = \begin{cases} 1 & \text{, iff } S_{t-1} = j \text{ and } S_t = k \\ 0 & \text{, otherwise.} \end{cases}$$

- Maximize the complete-data log-likelihood (CDLL):

$$\log\left(\mathbb{P}(\mathbf{x}^{(T)}, \mathbf{s}^{(T)})\right) = \underbrace{\sum_{j=1}^{N} u_j(1)\log \pi_j}_{\text{term 1}} + \underbrace{\sum_{j=1}^{N}\sum_{k=1}^{N}\left(\sum_{t=2}^{T} v_{jk}(t)\right)\log \gamma_{jk}}_{\text{term 2}}$$
$$+ \underbrace{\sum_{j=1}^{N}\sum_{t=1}^{T} u_j(t)\log p_j(x_t)}_{\text{term 3}}.$$

Prediction and Decoding

- h-step forecast distribution:

$$\mathbb{P}(X_{T+h} = x \mid \mathbf{X}^{(T)} = \mathbf{x}^{(T)}) = \frac{\mathbb{P}(\mathbf{X}^{(T)} = \mathbf{x}^{(T)}, X_{T+h} = x)}{\mathbb{P}(\mathbf{X}^{(T)} = \mathbf{x}^{(T)})} = \frac{\boldsymbol{\alpha}_T \boldsymbol{\Gamma}^h \mathbf{P}(x) \mathbf{1}'}{\boldsymbol{\alpha}_T \mathbf{1}'}.$$

- Local decoding:

$$S_t^* = \operatorname*{argmax}_{j=1,2,\ldots,N} \mathbb{P}(S_t = j \mid \mathbf{X}^{(T)} = \mathbf{x}^{(T)}).$$

- Global decoding:

$$\xi_{1i} = \mathbb{P}(S_1 = i, X_1 = x_1) = \pi_i p_i(x_1),$$

$$\xi_{ti} = \max_{s_1, s_2, \ldots, s_{t-1}} \mathbb{P}(\mathbf{S}^{(t-1)} = \mathbf{s}^{(t-1)}, S_t = i, \mathbf{X}^{(T)} = \mathbf{x}^{(T)}),$$

$$S_T = \operatorname*{argmax}_{i=1,2,\ldots,N} \xi_{Ti},$$

$$S_t = \operatorname*{argmax}_{i=1,2,\ldots,N} (\xi_{ti} \gamma_{i, S_{t+1}}).$$

Stock Return Series Prediction System

System Overview

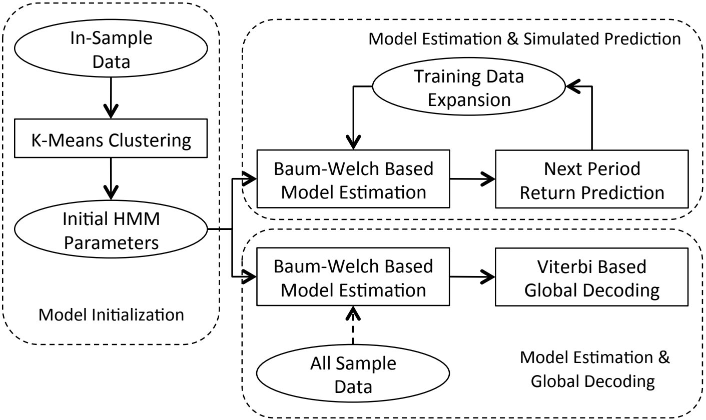


Figure 2: Overview of the stock return series prediction system

Model Initialization

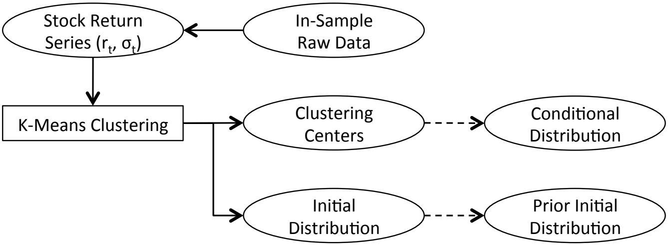


Figure 3: Model initialization module

- Transform the raw data into specified data structure (data tidying).
- Pre-process the raw data for observed variable series as K-Means input.

EM-Based Model Estimation

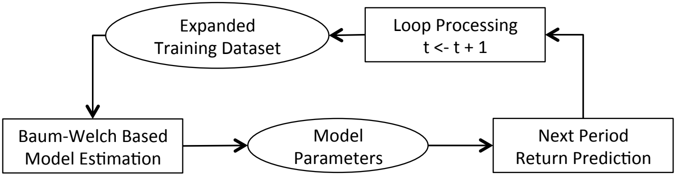


Figure 4: EM based model estimation procedure

- Carry out model estimation with EM algorithm (Baum-Welch).
- Calibrated parameters: hidden states initial distribution $\boldsymbol{\pi}$, hidden states final distribution $\boldsymbol{\delta}$, state transition matrix $\boldsymbol{\Gamma}$, conditional distribution parameters $(\boldsymbol{\mu}, \boldsymbol{\sigma})$.
- The model estimation procedure is *adaptive*, i.e. it incorporates the new historical return information when the loop processes.

Prediction

- Two different ways to predict the *price* of the next period:

$$\text{static: } \hat{P}_t = P_0 e^{\sum_{i=1}^{t} \hat{r}_i},$$
$$\text{adaptive: } \hat{P}_t = P_{t-1} e^{\hat{r}_t},$$

  where P_{t-1} is the real price at time $t-1$ and P_0 is the price of the first day when the prediction procedure begins.

- The first prediction procedure only considers the information of the returns, and thus the errors of predictions accumulate and are reflected on the predicted price.
- The second (so-called adaptive prediction) procedure considers both the return and the *price level* at $t-1$ to predict the price at t.

Global Decoding

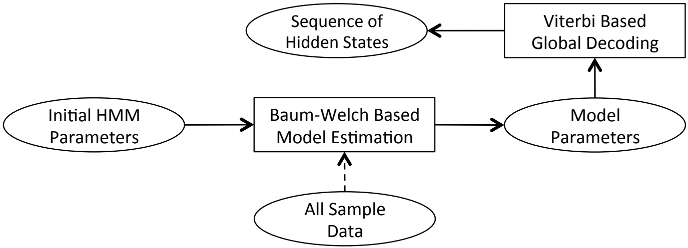


Figure 5: Global decoding module

- Independent with the simulated prediction module.
- Conduct Viterbi algorithm based global decoding to deduce the most probable sequence of hidden states during the observation period.

Empirical Analysis

Settings

- Stock returns conditional on the market states are normally distributed.
- Both Chinese and U.S. arkets have and only have three hidden states, namely bear, intermediate and bull:
  1. **Bear** means relatively low returns with high volatilities.
  2. **Intermediate** means returns and volatilities bot close to zero.
  3. **Bull** means relatively high returns with high volatilities.

  The number of hidden states is now artificially given and will be analyzed and validated later.
- Some conjectures:
  1. Time lag effects exist and worsen due to too much weight of outdated information.
  2. Results may vary with observation periods and frequencies.

Chinese CSI 300 Index daily return series

Figure 6: CSI 300 historical prices

- Data range from Mar. 5^{th}, 2013 to Mar 3^{rd}, 2016, 729 prices within three years in total.

Chinese CSI 300 Index daily return series

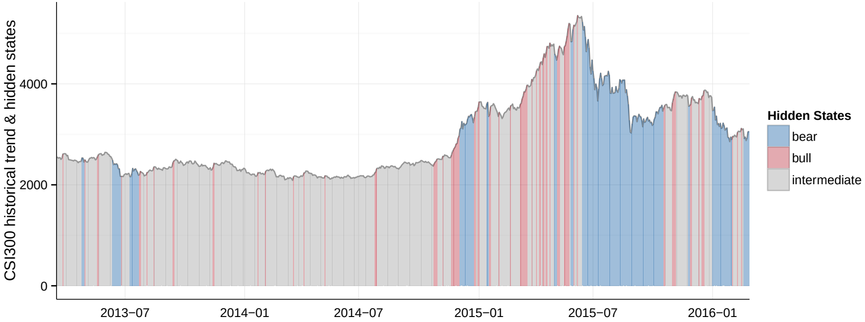


Figure 7: CSI 300 in-sample trend and states sequence

Chinese CSI 300 Index daily return series

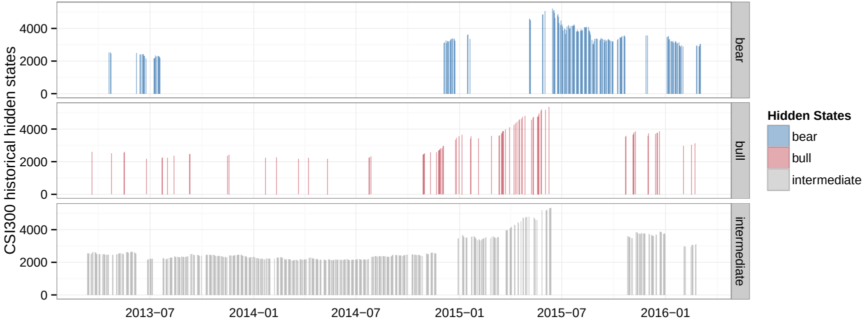


Figure 8: CSI 300 historical hidden states

Chinese CSI 300 Index daily return series

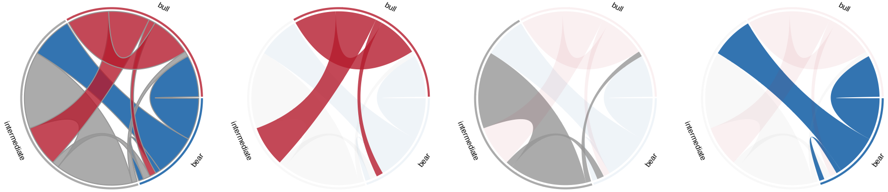


Figure 9: State transition for CSI 300 in-sample data

- Colored chords start from the (same) color to another color represent the transition probabilites.
- The thickness of the chords represent the relative relationships among the probabilities.

Chinese CSI 300 Index daily return series

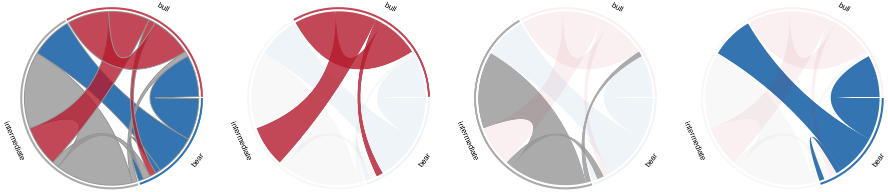


Figure 9: State transition for CSI 300 in-sample data

Example (Part 1 & 2 of Fig. 9)

There is a red chord starting from bull (red) and ending with intermediate (grey). It represents the probability that the bull transits into the intermediate. Red to grey is the thickest, meaning it is most likely for bull to go into the intermediate.

Chinese CSI 300 Index daily return series

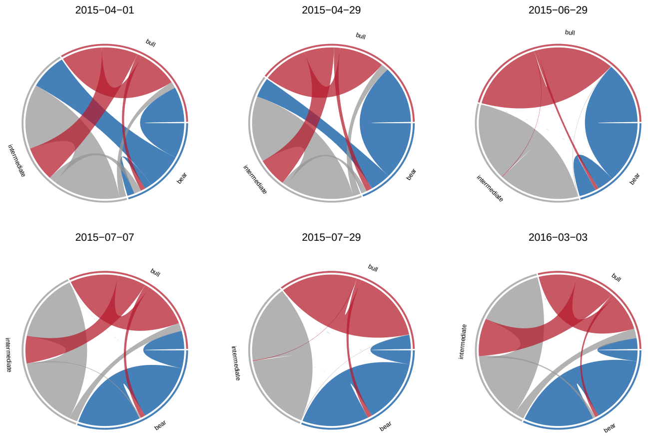


Figure 10: Typical state transition matrices during the entire period

Chinese CSI 300 Index daily return series

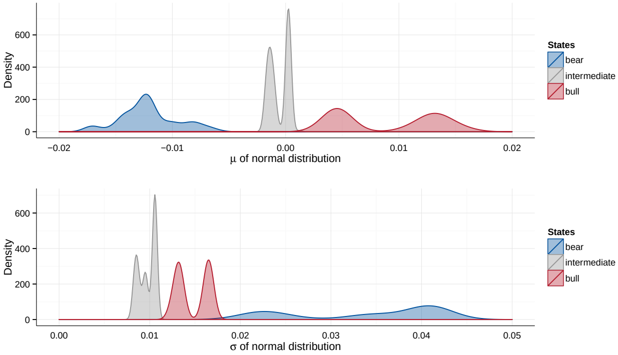


Figure 11: Density distribution of conditional distribution parameters

- Conditional distribution parameters change with time goes by.

Chinese CSI 300 Index daily return series

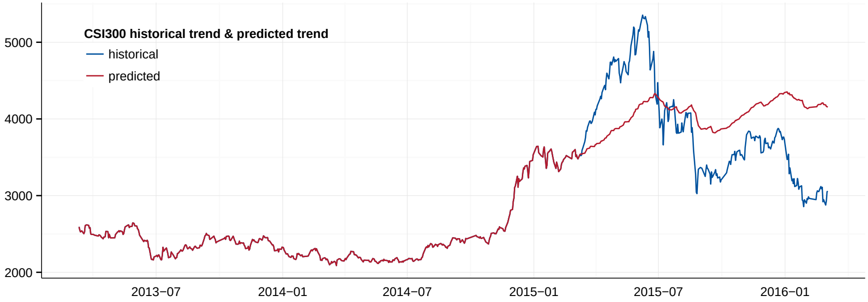


Figure 12: CSI 300 simulated prediction result

- Errors accumulate for long-period static predictions and time lag effects exist.

Chinese CSI 300 Index daily return series

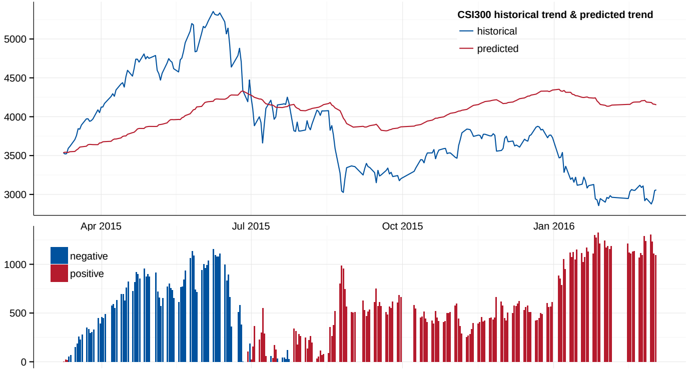


Figure 13: CSI 300 simulated prediction result out-of-sample part

Chinese CSI 300 Index daily return series

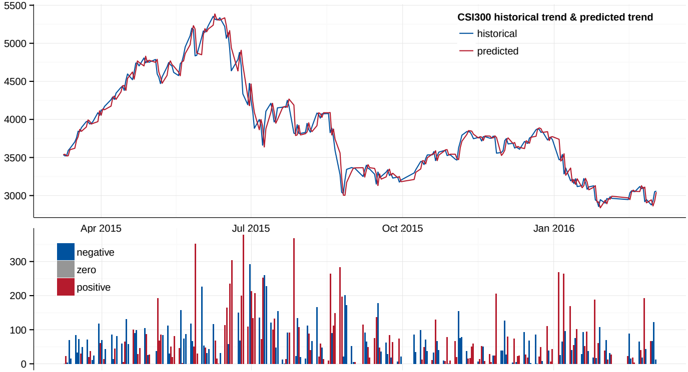


Figure 14: CSI 300 adaptive predictions

Chinese CSI 300 Index daily return series

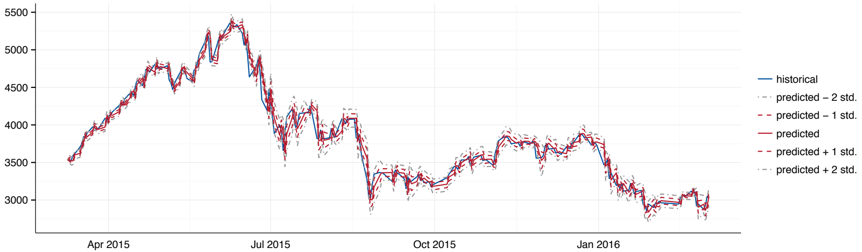
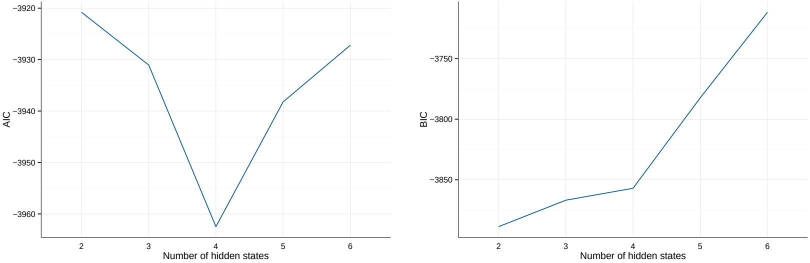


Figure 15: CSI 300 adaptive predictions with confidence interval

- All actual historical returns are within two standard deviations from the predicted return, and thus the adaptive predicted price level.

The Number of Hidden States



(a) AIC vs. N

(b) BIC vs. N

Figure 16: Goodness of fit for different number of hidden states

- Four is optimal from AIC and two from BIC, so three is reasonable.

Prediction Results Summary

Table 1: Brief description of all prediction results

Target Index	Data Frequency	Time Period	Data Length	Win Ratio
S&P 500	daily	2013∼2016	757	50.4%
CSI 300	daily	2013∼2016	729	50.2%
CSI 300	60min	2013∼2014	968	51.7%
CSI 300	60min	2014∼2015	976	55.5%
CSI 300	60min	2015∼2016	972	50.5%
CSI 300	10min	2013∼2014	5808	51.7%
CSI 300	10min	2014∼2015	5856	57.5%
CSI 300	10min	2015∼2016	5832	52.0%

Summary

Conclusion

For the System

- The system is complete, encapsulated and user-friendly.
- The system performs well in adaptive predictions, all actual returns fall in the 95.45% confidence interval.
- Time lag effects exist, so sample reweighting is necessary to improve the performance.

For Empirical Analyses

- Data populations with longer observation periods tend to provide more accurate hidden states anlysis results then short-term data.
- Data populations with higher observation frequencies tend to outperform those with lower frequencies w.r.t. prediction.

Future Work

Time Lag Effects

- **Rolling window method**, more popular in the industry, the size of the window should be optimized.
- **Exponential weighted expectation maximization (EWEM) algorithm**, proposed in [Zhang(2004)], more complex to implement.

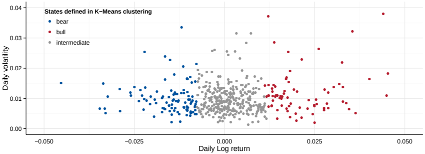
Particle Filter

- Standard tool for HMM estimation when the latent variable is continuous, one of the most famous cases being Kalman filter.
- PF can incorporates economic meanings into values of the latent variable while currently we only consider the hidden states as categorical.

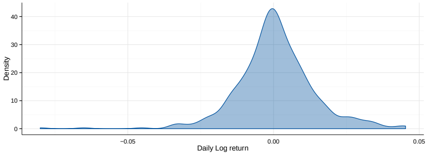
Appendix

Full Presentation of CSI 300 Results Visualization

(a) CSI 300 daily index



(b) Daily returns and K-Means states



(c) Daily returns distribution

Figure 17: Data description

Full Presentation of CSI 300 Results Visualization

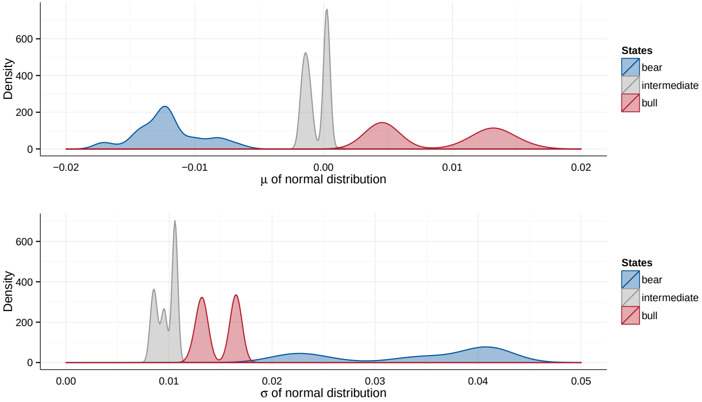
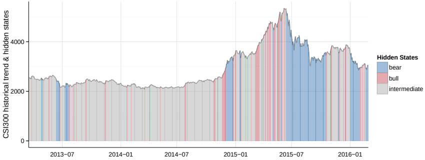
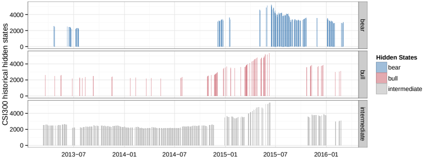


Figure 18: Density distribution of conditional distribution parameters

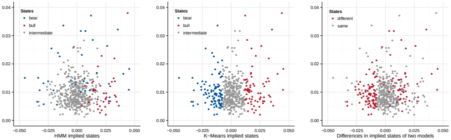
Full Presentation of CSI 300 Results Visualization



(a) Historical index and states



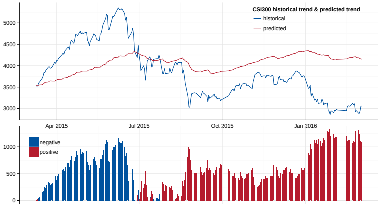
(b) Hidden states facets



(c) HMM and K-Means states comparison

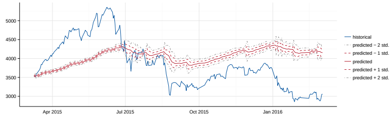
Figure 19: Historical hidden states results

Full Presentation of CSI 300 Results Visualization

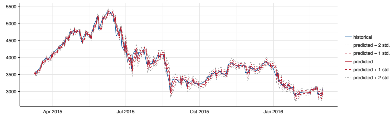


(a) Simulated prediction result

(b) Adaptive prediction result



(c) Simulated prediction with confidence band



(d) Adaptive prediction with confidence band

Figure 20: Simulated prediction results

References I

Hasselblad V.
Estimation of Finite Mixtures of Distributions from the Exponential Family[J].
Journal of the American Statistical Association, 1969, 64(328):1459–1471.

Behboodian J.
On the Modes of a Mixture of Two Normal Distributions[J].
Technometrics, 1970, 12(1):131–139.

Forgy E W.
Cluster analysis of multivariate data: efficiency versus interpretability of classifications[J].
Biometrics, 1965, 21:768–769.

References II

Zucchini W, MacDonald I L.
Hidden Markov models for time series[M],
Monographs on Statistics and Applied Probability, vol. 110.[S.l.]: CRC Press, Boca Raton, FL, 2009.

Baum L E, Petrie T.
Statistical inference for probabilistic functions of finite state Markov chains[J].
Annals of Mathematical Statistics, 1966, 37(6):1554–1563.

Baum L E, Eagon J A.
An inequality with applications to statistical estimation for probabilistic functions of Markov processes and to a model for ecology[J].
Bulletin of the American Mathematical Society, 1967, 73(3):360–363.

References III

Baum L E, Petrie T, Soules G, et al.
A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains[J].
Annals of Mathematical Statistics, 1970, 41(1):164–171.

Zhang Y.
Prediction of financial time series with Hidden Markov Models[J].
2004.

Questions?

Thanks for your time!