Applied Mathematics
&
Statistics

Technical Report - Fall 2022

Statistical Inference of Hidden Markov Models on High Frequency Quote Data

Jason Bohne

Contents

1 Introduction

Hidden Markov models first appeared in the statistics literature in the 1960s, and have gained much popularity across various disciplines and applications since their original inception. In a general sense, Hidden Markov models are discrete-time stochastic models with hidden states that determine the probability distribution of the observed outcomes. In practice, it is common to assume the underlying distribution of observations across distinct states is of the same functional form with different parametrizations.

The vast majority of literature on the applications of Hidden Markov models focuses on speech recognition, as in [Rabiner, 1989], and biological sequencing in the canonical text by [Birney, 2001]. In recent years, there has been an increase in applications of such models to topics in quantitative finance with [Zhang et al., 2018] training their model for low-frequency price trend prediction and [Rossi and Gallo, 2006] considering volatility estimation. While applications of continuous-time stochastic models in high-frequency finance are widespread, see [Cont and de Larrard, 2013] and [Cont et al., 2010]; the use of discrete-time stochastic models, specifically Hidden Markov Models, has been relatively unexplored. One paper that does consider this application is the work by [Sandoval and Hernández, 2015] who train a Hierarchical Hidden Markov model using high-frequency trade and quote data to model short-term price trends.

The report will proceed as follows. In section 2.1 an overview will be given of Hidden Markov models, a common stochastic model, and the representation of their likelihood function. From there, methodologies for statistical inference are provided notably the Baum-Welch algorithm in section 3.1 and the Viterbi algorithm in section 3.2. Section 4 discusses the preprocessing techniques utilized to prepare high-frequency quote data for the analysis. Focusing on the optimization problem, section 5 includes the formulation of the objective and constraints along with the numerical solutions when solving the problem using PSG and Hmmlearn. Finally, the main ideas and thoughts will be shared in section 6, the conclusion. The appendix includes access to an implementation of the method in python along with reference data.

2 Hidden Markov Model

2.1 Model Specification

Hidden Markov models are state-space models, characterized by the unknown discrete states $(X_1 \dots X_N)$ responsible for the underlying distributions of the observations $(Y_1 \dots Y_n)$. Moreover, it is assumed for X to follow a first-order Markov chain such that

$$P(X_{t+1} = S_j | X_0 = S_0 \dots X_t = S_i) = P(X_{t+1} = S_j | X_t = S_i)$$

Therefore our model can be specified entirely by the following given the hyperparameter k, representing number of states, and n observed samples of (x_t, y_t) where A is the state transition matrix, B_j is the underlying probability density of state S_j, and π is our initial state distribution matrix.

$$A = P(X_t | X_{t=1} = S_j)$$

$$B = P(y_t | X_t = S_j),\ \pi = P(X_0)$$

2.2 Likelihood Function

As typical in maximum likelihood estimation, the likelihood function $l(\theta) = P(O|\theta)$ of the model given the observations O is required for statistical inference. First derived in [Rabiner, 1989], the conditional probability of transitioning between state S_i and state S_j at time t as the following:

$$\varepsilon_t(i, j) = P(X_t = S_i, X_{t+1} = S_j | O, \theta)$$

The conditional probability of being in state S_j can then be computed as

$$\gamma_j(t) = \sum_{i=1}^{k} \varepsilon_t(i, j)$$

With the expected number of occurrences of S_j across all hidden states as

$$E(I_{S_i = S_j}) = \sum_{t=1}^{n} \gamma_j(t)$$

Note the quantities above are commonly used to determine the update procedure when determining maximum likelihood estimates for the parameters $\hat{A},\ \hat{B_j},\ \hat{\pi}$.

3 Statistical Inference

3.1 Baum-Welch Algorithm

Credited for developing the underlying theory, [Baum and Petrie, 1966] introduced a maximum likelihood estimation procedure, termed the Baum-Welch algorithm, to estimate the unknown parameters θ_0 of a Hidden Markov model. Here $\hat{\theta}$ is composed of the transition matrix $\hat{A}$, distribution of the hidden states $\hat{B}_j$, and initial conditions $\hat{\pi}$ within a discrete Markov Chain. Computationally, the algorithm falls within the class of Expectation-Maximization algorithms, which iteratively updates the estimate of $\hat{\theta}$ until $\hat{\theta}$ is a local maximum of the likelihood function. Using the results of likelihood function, [Rabiner, 1989] provides the following reestimation rules for updating the transition matrix, probability distribution of each state, and initial state distribution.

$$\hat{\pi} = \gamma_1(i)$$

$$\hat{A} = \sum_{t=1}^{n-1} \frac{\varepsilon_t(i,j)}{\gamma_t(i)}$$

$$\hat{B}_j(k) = \frac{\sum_{t=1}^{n-1} \gamma_t(j) I_{O_t=k}}{\sum_{t=1}^{n-1} \gamma_t(j)}$$

As traditional in iterative algorithms; the solution vector $\hat{\theta}$ is updated in each iteration. While the convergence of the inference procedure is not guaranteed, it is known the likelihood function $l(\hat{\theta}) = (\hat{A},\ \hat{B}_j, \hat{\pi})$ will be monotonically increasing across iterations.

3.2 Viterbi Algorithm

Originally proposed in [Viterbi, 1967] the Viterbi algorithm utilizes dynamic programming to recursively determine the optimal kth subsequence of hidden states $\forall k \in \{1, \ldots, n\}$ conditional on the observations. Given the original model likelihood $l(\hat{\theta})$ the Viterbi algorithm partitions the objective into a sequence of recursive functions as below:

$$l(\hat{\theta}) = l_1(\hat{\theta}) + l_1^2(\hat{\theta}) + \ldots + l_{n-1}^n(\hat{\theta})$$

As standard in dynamic programming frameworks, the original problem is recursively traced backward to the initial state where the optimal hidden state can be determined. This solution is then iteratively applied to solve the problem forwards in time.

4 Data Preparation

When working with big data, it is common to perform preprocessing and feature generation on the raw data before fitting statistical models. The original dataset consists of 1.8 million Apple, AAPL, top of the book quotes throughout the month of January 2020, across the 13 stock exchanges operating for U.S. equities. In particular, quote data refers to top-of-the-book limit orders, specified by the bid and offer price and size denoted respectively as BP_i, OP_i, BS_i, OS_i.

4.1 Preprocessing

Steps were taken for preprocessing the original quote data

- Sort and index quotes by participant timestamp
- Remove quotes outside of market hours
- Remove quotes with zero price
- Remove quotes of inverted spread
- Remove quotes of zero posted size

4.2 Feature Generation

Features relevant to the limit order book were specifically chosen prior to the analysis, with the bid and offer size coming direct from the original data. Transformations on the price and size features generated the order book imbalance, OB_i, and spread, S_i, variables. Log transformations were applied to the entire feature set to lessen the effect of outliers. Raw features were aggregated into one-second bars by taking the sample mean.

$$F = \{BS_i, OS_i, OB_i = \frac{OS_i}{BS_i}, S_i = OP_i - BP_i\}$$

Histograms are commonly used to visualize the empirical distribution of a feature. Note below, all of the empirical distributions portray skewness with varying similarities to a normal distribution. Time-dependent plots are also included to display the evolution of the features across a sample date.

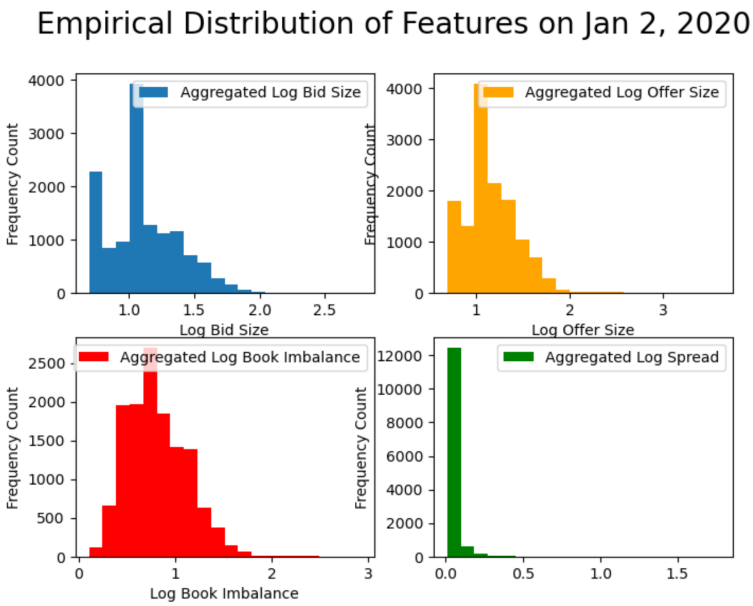

Figure 1: Empirical Distribution of Features

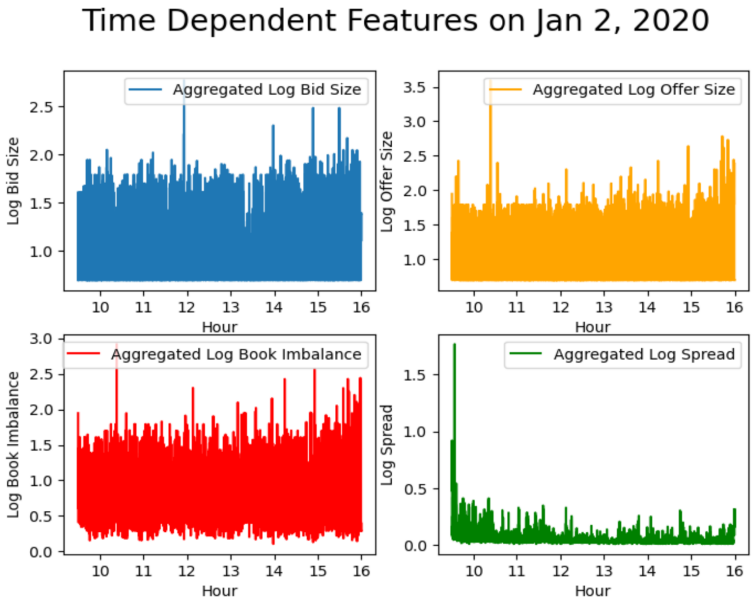

Figure 2: Time Dependent Plot of Features

5 Numerical Results

5.1 Setup

The underlying state distribution X_i of each feature is assumed to follow a Markov Chain which is composed at any point in time by one of two possible normal distributions denoted respectively as $\mathcal{N}(\mu_1, \sigma_1^2)$ and $\mathcal{N}(\mu_2, \sigma_2^2)$. The inference problem of determining the most likely parameterizations and transition matrix given the observed sequence can be solved as an optimization problem of the model likelihood $l(\hat{\theta})$; both using constrained optimization starting with the Baum-Welch algorithm in PSG by [American Optimal Decisions,] and the Viterbi algorithm in Hmmlearn by [Developers,].

For each feature a Hidden Markov model will be fitted on a single day of high-frequency quote data in the dataset, using both implementations. Assuming each day to be *i.i.d.* the procedure will be repeated across the full month of data to improve the sample size of the parameter estimates. Sample means and standard deviations of the parameter estimates can be found below; along with p-values corresponding to a two-sample t-test, as discussed in [Snedecor and Cochran, 1989], with the hypothesis that the population means of the underlying distributions are equivalent.

5.2 PSG Results

The multistage optimization routine within PSG utilizes the Baum-Welch algorithm as discussed in 3.1 to determine the initial starting point of the likelihood function $l(\hat{\theta})$. From there constraints on the parameters a_{ij} of the transition matrix and π for the initial state distribution are automatically generated to respect conditions for valid probability measures as noted in [Uryasev,]. Specifically where

$$\sum_{i=1}^{k} p_i = 1 \ \forall p_i \geq 0, \ \sum_{i=1}^{k} \sum_{j=1}^{k} a_{ij} = 1 \ \forall a_{ij} \geq 0$$

Features	a_{11}	a_{12}	a_{21}	a_{22}	mu_1	si_1	mu_2	si_2
BS	0.7756	0.2243	0.0573	0.9426	1.0700	0.2195	1.4669	0.3708
OS_i	0.7998	0.2001	0.1048	0.8951	1.1015	0.2153	1.4351	0.3566
OB_i	0.8594	0.1405	0.1046	0.8953	0.6567	0.2092	1.0311	0.3188
S_i	0.8062	0.1937	0.1335	0.8664	0.0606	0.0208	0.1818	0.1225

Table 1: Sample Mean of PSG Parameter Estimates

Features	a_{11}	a_{12}	a_{21}	a_{22}	mu_1	si_1	mu_2	si_2
BS	0.2156	0.2156	0.0412	0.0412	0.0982	0.0236	0.2063	0.1694
OS_i	0.2660	0.2660	0.2182	0.2182	0.1072	0.0609	0.2886	0.1951
OB_i	0.0842	0.0842	0.0663	0.0663	0.0668	0.0219	0.1297	0.1100
S_i	0.1056	0.1056	0.0988	0.0988	0.0671	0.0340	0.1334	0.0639

Table 2: Sample Standard Deviation of PSG Parameter Estimates

Features	p
BS_i	0.2836
OS_i	0.4200
OB_i	0.2452
S_i	0.2221

Table 3: Average p-values for two sample t-test on the population mean

5.3 Hmmlearn Results

Hmmlearn allows for flexibility when choosing the algorithm to solve the inference problem. For all of the numerical results, the Viterbi algorithm was chosen as discussed in 3.2. Additional parameters specified were the covariance threshold, tolerance, and max iterations, which were set to $10e^{-4}$, $10e^{-8}$ and $10e^{3}$ respectively.

Features	a_{11}	a_{12}	a_{21}	a_{22}	mu_1	si_1	mu_2	si_2
BS_i	0.9201	0.0798	0.1131	0.8868	1.0225	0.0478	1.3749	0.1196
OS_i	0.6505	0.3494	0.3655	0.6344	1.0756	0.0332	1.262011	0.0992
OB_i	0.8429	0.1570	0.1583	0.8416	0.6687	0.0473	1.0231	0.1143
S_i	0.8857	0.1142	0.2120	0.7879	0.0606	0.0015	0.1822	0.0189

Table 4: Sample Mean of Hmmlearn Parameter Estimates

Features	a_{11}	a_{12}	a_{21}	a_{22}	mu_1	si_1	mu_2	si_2
BS_i	0.0893	0.0893	0.1307	0.1307	0.0916	0.0180	0.2086	0.1454
OS_i	0.3884	0.3884	0.3933	0.3933	0.0845	0.0278	0.0419	0.1108
OB_i	0.1705	0.1705	0.1811	0.1811	0.0738	0.0166	0.1376	0.1170
S_i	0.0764	0.0764	0.1097	0.1097	0.0671	0.0060	0.1333	0.0225

Table 5: Sample Standard Deviation of Hmmlearn Parameter Estimates

Features	p
BS_i	0.1276
OS_i	0.0280
OB_i	0.0500
S_i	0.0003

Table 6: Average p-values for two sample t-test on the population mean

5.4 Algorithm Comparison

It is worth examining the optimal parameterization of the underlying distributions for a given feature as one disadvantage of Hidden Markov models is their inability to differentiate whether or not a feature truly follows a state-space model of k distinct states. Performing a two-sample t-test on the estimates of the population mean for each distribution, one can statistically quantify the probability that the two hidden states actually follow the same underlying distribution.

When the problem is solved via constrained optimization, the average p-values are nonzero and the hypothesis that the true distributions are equivalent would not be rejected at the 10% significance level. On the contrary, when solving with the Viterbi algorithm, the average of most p-values is near zero, which would allow for the rejection of the hypothesis at a 10% significance level. It is important to realize the solution in the former would not support the argument that the true process of the features follows a state-space model alternating between two distinct states, however, the solution in the latter would. One reason causing this inconsistency could be in the estimates for the standard deviation between the implementations. When solving the problem via constrained optimization the estimates for the standard deviation of the hidden states are almost always much greater in magnitude than the estimates derived with the Viterbi algorithm, leading to greater uncertainty.

To compare the parameterization results between approaches, a two-sample t-test is performed once again. However this time it tests whether the underlying distribution $\mathcal{N}(\mu_i, \sigma_i^2)$ of each feature is equivalent across implementations. Examining the p-values would lead to not rejecting the hypothesis that the underlying distributions are equal at a 10% significance level. The implication of this supports the conclusion that both implementations parameterize the underlying distribution similarly when solving the statistical inference problem of fitting a Hidden Markov model with Gaussian hidden states to a sequence of observations.

Features	p_1	p_2
BS_i	0.4092	0.4100
OS_i	0.3982	0.4034
OB_i	0.5034	0.4864
S_i	0.5124	0.7450

Table 7: Average p-values for two sample t-test across implementations

6 Conclusion

In sum, this report focuses on statistical inference methods for Hidden Markov models and provides an application to high-frequency finance. Given the general stochastic model formulation outlined in section 2.1, methods for inference, specifically the Baum-Welch algorithm in section 3.1 and the Viterbi algorithm in section 3.2, are discussed. From there, standard techniques for cleaning the original data and feature generation are included in section 4.1 and section 4.2 respectively. Transitioning to the optimization problem, the objective and constraints are outlined and solved with PSG and Hmmlearn; to estimate the parameters for the models trained on each individual feature in section 5.

While there is variation in the results of the optimal parameters of the underlying distributions between implementations; both methods result in an estimate of the true distribution supported by the two-sample t-test. This is a major advantage when performing statistical inference, as it allows for greater flexibility in algorithm choice to solve the optimization problem. That being said, as mentioned in section 5 one disadvantage when fitting Hidden Markov models is the ambiguity surrounding the choice of hyperparameters in the model specification. Further investigation is still required to determine whether a Hidden Markov model of two normally distributed hidden states is optimal for the features chosen in this analysis. Improvements can also be made by the formulation and inference of a Hidden Markov model on multivariate observations which can be the composition of the features mentioned above.

7 Appendix

7.1 Code Access

The full repository of all scripts and data utilized in this project can be accessed on **GitHub**.

7.2 Top of the Book Data Sample

Top of the book limit orders are the quotes corresponding to the highest bid and the lowest offer on an exchange at any point in time. Quotes are updated either if a higher bid/lower offer is quoted or if the original liquidity on a certain level is taken by market participants. In addition to the price and size attributes, each quote includes the source of exchange in addition to the national best-bid-and-offer; the highest bid, and the lowest offer across all exchanges. It is worth pointing out that the quote data is indexed based on the participant timestamp instead of the SIP timestamp here as inaccuracies in labeling have been found by [Schwenk-Nebbe, 2022] in the latter.

	SIP_Timestamp	Exchange	Symbol	Bid_Price	Bid_Size	Offer_Price	Offer_Size	Quote_Condition	Sequence_Number	Source_Of_Quote	...	Best_Bid_Price	Best_Bid_Size	Best_Offer_Exchange	Best_Offer_Price	Best_Offer_Size	L
2020-01-02 09:30:00.134062	2020-01-02 09:30:00.134429	P	AAPL	296.09	1.0	296.29	1.0	R	262393	N	...	296.24	2.0	P	296.29	1.0	
2020-01-02 09:30:00.134336	2020-01-02 09:30:00.134554	K	AAPL	296.21	1.0	296.39	1.0	R	262394	N	...	296.21	1.0	P	296.29	1.0	
2020-01-02 09:30:00.134532	2020-01-02 09:30:00.134742	K	AAPL	296.10	1.0	296.36	2.0	R	262401	N	...	296.10	1.0	P	296.29	1.0	
2020-01-02 09:30:00.136081	2020-01-02 09:30:00.136273	K	AAPL	296.10	1.0	296.29	1.0	R	262424	N	...	296.10	1.0	P	296.29	1.0	
2020-01-02 09:30:00.234474	2020-01-02 09:30:00.234700	K	AAPL	296.11	1.0	296.39	1.0	R	263840	N	...	296.11	1.0	P	296.29	1.0	
2020-01-02 09:30:00.330155	2020-01-02 09:30:00.330381	K	AAPL	296.10	2.0	296.39	1.0	R	265507	N	...	296.10	2.0	P	296.29	1.0	
2020-01-02 09:30:00.346590	2020-01-02 09:30:00.346608	Q	AAPL	296.11	1.0	296.40	1.0	R	265730	N	...	296.11	1.0	P	296.29	1.0	
2020-01-02 09:30:00.548226	2020-01-02 09:30:00.548308	Q	AAPL	296.11	1.0	296.41	4.0	R	268373	N	...	296.11	1.0	P	296.29	1.0	
2020-01-02 09:30:00.638131	2020-01-02 09:30:00.638149	Q	AAPL	296.11	1.0	296.27	1.0	R	269236	N	...	296.11	1.0	Q	296.27	1.0	
2020-01-02 09:30:00.638137	2020-01-02 09:30:00.638153	Q	AAPL	296.11	1.0	296.25	2.0	R	269237	N	...	296.11	1.0	Q	296.25	2.0	
2020-01-02 09:30:00.638278	2020-01-02 09:30:00.638499	K	AAPL	296.11	1.0	296.25	2.0	R	269246	N	...	296.11	1.0	Q	296.25	2.0	
2020-01-02 09:30:00.668135	2020-01-02 09:30:00.668152	Q	AAPL	296.11	1.0	296.41	4.0	R	269579	N	...	296.11	1.0	Z	296.25	2.0	
2020-01-02 09:30:00.668239	2020-01-02 09:30:00.668427	Z	AAPL	295.91	1.0	296.40	1.0	R	269587	N	...	296.11	1.0	K	296.25	2.0	
2020-01-02 09:30:00.668244	2020-01-02 09:30:00.668462	K	AAPL	296.10	2.0	296.28	1.0	R	269589	N	...	296.11	1.0	K	296.28	1.0	
2020-01-02 09:30:00.701300	2020-01-02 09:30:00.708643	Q	AAPL	296.25	26.0	296.30	5.0	R	270061	N	...	296.25	26.0	K	296.28	1.0	
2020-01-02 09:30:00.720653	2020-01-02 09:30:00.720877	K	AAPL	296.10	2.0	296.39	1.0	R	270223	N	...	296.25	26.0	Q	296.28	1.0	
2020-01-02 09:30:00.724376	2020-01-02 09:30:00.724394	X	AAPL	295.44	2.0	296.27	1.0	R	270247	N	...	296.25	26.0	X	296.27	1.0	
2020-01-02 09:30:00.724398	2020-01-02 09:30:00.724415	X	AAPL	295.44	2.0	297.06	3.0	R	270248	N	...	296.25	26.0	Q	296.28	1.0	
2020-01-02 09:30:00.758366	2020-01-02 09:30:00.758383	Q	AAPL	296.25	26.0	296.30	5.0	R	270638	N	...	296.25	26.0	B	296.28	1.0	

Figure 3: First 25 Quotes for AAPL on Jan. 02, 2020

References

[American Optimal Decisions,] American Optimal Decisions, I. Portfolio safeguard.

[Baum and Petrie, 1966] Baum, L. E. and Petrie, T. (1966). Statistical Inference for Probabilistic Functions of Finite State Markov Chains. *The Annals of Mathematical Statistics*, 37(6):1554 – 1563.

[Birney, 2001] Birney, E. (2001). Hidden markov models in biological sequence analysis. *IBM Journal of Research and Development*, 45(3.4):449–454.

[Cont and de Larrard, 2013] Cont, R. and de Larrard, A. (2013). Price dynamics in a markovian limit order market. *Econometrics: Applied Econometrics & Modeling eJournal*.

[Cont et al., 2010] Cont, R., Stoikov, S., and Talreja, R. (2010). A stochastic model for order book dynamics. *Operations Research*, 58:549–563.

[Developers,] Developers, H. Unsupervised learning and inference of hidden markov models.

[Rabiner, 1989] Rabiner, L. (1989). A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286.

[Rossi and Gallo, 2006] Rossi, A. and Gallo, G. M. (2006). Volatility estimation via hidden markov models. *Journal of Empirical Finance*, 13(2):203–230.

[Sandoval and Hernández, 2015] Sandoval, J. and Hernández, G. (2015). Computational visual analysis of the order book dynamics for creating high-frequency foreign exchange trading strategies. *Procedia Computer Science*, 51:1593–1602. International Conference On Computational Science, ICCS 2015.

[Schwenk-Nebbe, 2022] Schwenk-Nebbe, S. (2022). The participant timestamp: Get the most out of taq data. Workingpaper.

[Snedecor and Cochran, 1989] Snedecor, G. W. and Cochran, W. G. (1989). *Statistical Methods*, volume 8.

[Uryasev,] Uryasev, S. Case study: Maximization of log-lokelihood in hidden markov model.

[Viterbi, 1967] Viterbi, A. (1967). Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13(2):260–269.

[Zhang et al., 2018] Zhang, M., Jiang, X., Fang, Z., Zeng, Y., and Xu, K. (2018). High-order hidden markov model for trend prediction in financial time series. *Physica A: Statistical Mechanics and its Applications*, 517.