

# Auto Regressive Dynamic Bayesian Network and Its Application in Stock Market Inference

Tiehang Duan

► **To cite this version:**

Tiehang Duan. Auto Regressive Dynamic Bayesian Network and Its Application in Stock Market Inference. 12th IFIP International Conference on Artificial Intelligence Applications and Innovations (AIAI), Sep 2016, Thessaloniki, Greece. pp.419-428, 10.1007/978-3-319-44944-9\_36 . hal-01557633

**HAL Id: hal-01557633**

**<https://hal.inria.fr/hal-01557633>**

Submitted on 6 Jul 2017

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



# Auto Regressive Dynamic Bayesian Network and Its Application in Stock Market Inference

Tiehang Duan

Department of Computer Science and Engineering  
The State University of New York at Buffalo  
Buffalo, NY 14226, United States  
`tiehangd@buffalo.edu`

**Abstract.** In this paper, auto regression between neighboring observed variables is added to Dynamic Bayesian Network(DBN), forming the Auto Regressive Dynamic Bayesian Network(AR-DBN). The detailed mechanism of AR-DBN is specified and inference method is proposed. We take stock market index inference as example and demonstrate the strength of AR-DBN in latent variable inference tasks. Comprehensive experiments are performed on S&P 500 index. The results show the AR-DBN model is capable to infer the market index and aid the prediction of stock price fluctuation.

**Keywords:** Auto Regressive Dynamic Bayesian Network(AR-DBN), Expectation-Maximization(EM), Kullback–Leibler Divergence, Sum-Product Algorithm, Belief Propagation

## 1 Introduction

Dynamic Bayesian Network(DBN) uses directed graph to model the time dependent relationship in the probabilistic network. The method achieved wide application in gesture recognition [20] [17], acoustic recognition [3] [22], image segmentation [9] and 3D reconstruction [6]. The temporal evolving feature also makes the model suitable to model the stock market [7]. The classic DBN model assumes the observed variable only depend on latent variables, and we know for the instance of stock market, auto regression widely exists in neighboring observed stock prices due to the momentum of market atmosphere. So we introduce explicit auto regressive dependencies between adjacent observed variables in DBN, forming the Auto Regressive Dynamic Bayesian Network(AR-DBN).

In section 2, we have a brief review of previous work in related fields. In section 3, the structure of the network is formed and probability factors are derived. In section 4, we specify the parameter estimation of AR-DBN and experiments are conducted in section 5. Conclusion is reached in section 6.

## 2 Related Work

Researchers have been using Dynamic Bayesian Networks(DBN) to model the temporal evolution of stock market and other financial instruments [19]. In 2009,

Aditya Tayal utilized DBN to analyze the switching of regimes in high frequency stock trading [21]. In 2013, Zheng Li et al. used DBN to explore the dependence structure of elements that influence stock prices [10]. And in 2014, Jangmin O built a price trend model under the DBN framework [7]. Auto regression is an important factor that contribute to the fluctuation of stock prices and has been studied among researchers in financial mathematics [14] [12] [13]. Auto regressive relationships among adjacent observed variables is also used in Hidden Markov Models(HMM). In 2009, Matt Shannon et al. illustrated Auto Regressive Hidden Markov Model(AR-HMM) and used it for speech synthesis [18]. In 2010, Chris Barber et al. used AR-HMM to predict short horizon wind with incomplete data [2]. And in 2014, Bing Ai et al. estimated the smart building occupancy with the method [1]. To our knowledge, there haven't been previous investigation of auto regression applied to Dynamic Bayesian Network(DBN), and in this paper, we integrate the auto regressive property into DBN and forms the Auto Regressive Dynamic Bayesian Network(AR-DBN).

### 3 The Dynamic Bayesian Network for Stock Market Inference

The original DBN before integrating auto regression is shown in Fig. 2. For each time slice  $i$ , it includes  $m$  observed stock price variables  $Y_{1i}, \dots, Y_{mi}$  and hidden variable  $X_i$ . We denote the graph of each slice as  $B = (G, \theta)$ , where  $G$  is the structure of Bayesian Network(BN) for the slice, whose nodes corresponds to the variables and whose edges represent their conditional dependencies, and  $\theta$  represents the set of parameters encoding the conditional probabilities of each node variable given its parent [11]. The distribution is represented as CPD(conditional probabilistic distribution) [5]. In our case, as the observed variables are continuous, exact inference is achieved with sum product algorithm [4], which iterate between summation of belief in different states for each clique and combining the belief of neighboring cliques. After the end of each iteration, the marginal probability of each variable is inferred based on likelihood of the whole graph [15].

The likelihood of the graph in Fig. 2 is

$$\phi = P(X_1, Y_{11}, \dots, Y_{m1}, X_2, Y_{12}, \dots, Y_{m2}, \dots, X_n, Y_{1n}, \dots, Y_{mn}) \quad (1)$$

Define

$$\begin{aligned} \phi_1 &= P(X_1, Y_{11}, \dots, Y_{m1}) \\ \phi_2 &= P(X_1, Y_{11}, \dots, Y_{m1}, X_2, Y_{12}, \dots, Y_{m2}) \\ &\dots \\ \phi_n &= P(X_1, Y_{11}, \dots, Y_{m1}, X_2, Y_{12}, \dots, Y_{m2}, \dots, X_n, Y_{1n}, \dots, Y_{mn}) \\ \psi_n &= P(X_n, Y_{1n}, \dots, Y_{mn}) \\ \psi_{n-1} &= P(X_{n-1}, Y_{1(n-1)}, \dots, Y_{m(n-1)}, X_n, Y_{1n}, \dots, Y_{mn}) \end{aligned}$$

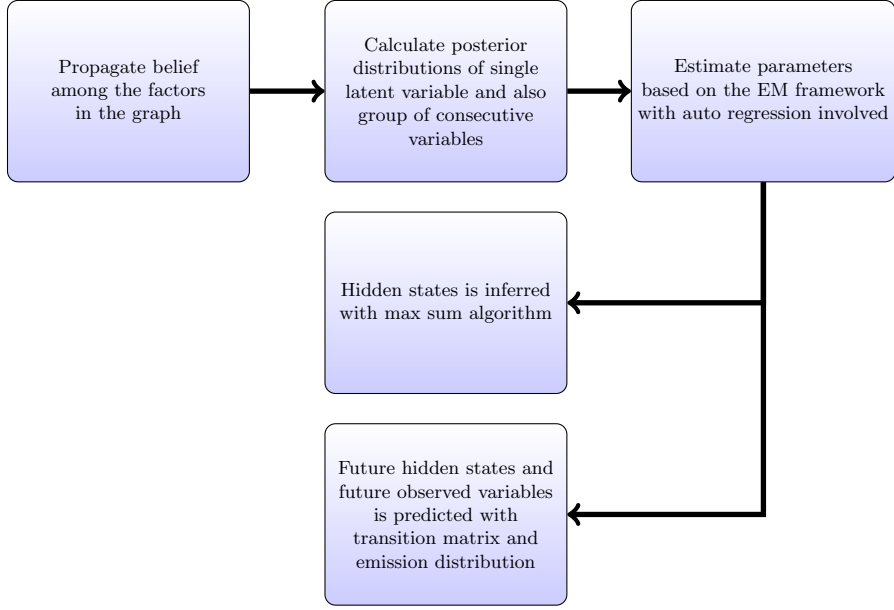


Fig. 1: Overview of Work Flow for AR-DBN model

.....

$$\psi_1 = P(X_1, Y_{11}, \dots, Y_{m1}, X_2, Y_{12}, \dots, Y_{m2}, \dots, X_n, Y_{1n}, \dots, Y_{mn})$$

also define

$$f(X_{i-1}, X_i) = P(X_i | X_{i-1})$$

$$f(X_i, Y_{1i}, \dots, Y_{mi}) = P(Y_{1i}, \dots, Y_{mi} | X_i)$$

Based on sum product algorithm, we have

$$\phi_i = \left( \sum_{x_{i-1}} \phi_{i-1} \times f(X_{i-1}, X_i) \right) \times f(X_i, Y_{1i}, \dots, Y_{mi}) \quad (2)$$

$$\psi_i = \left( \sum_{x_{i+1}} \psi_{i+1} \times f(X_i, X_{i+1}) \right) \times f(X_i, Y_{1i}, \dots, Y_{mi}) \quad (3)$$

And after the belief completes one round bidirectional propagation through the network, the  $\phi_i$  and  $\psi_i$  can be readily used to calculate the posterior distribution of each latent variable  $X_i$

$$P(X_i | Y) \propto \frac{\phi_i \times \psi_i}{f(X_i, Y_{1i}, \dots, Y_{mi})} \quad (4)$$

And we can similarly calculate the marginal probability of k consecutive hidden units  $X_i, X_{i+1}, \dots, X_{i+k}$

$$\begin{aligned}
 P(X_i, X_{i+1}, \dots, X_{i+k}|Y) &\propto \phi_i \times \psi_{i+k} \times f(X_i, X_{i+1}) \times \dots \times f(X_{i+k-1}, X_{i+k}) \\
 &\quad \times f(X_{i+1}, Y_{1(i+1)}, \dots, Y_{m(i+1)}) \times \dots \times f(X_{i+k-1}, Y_{1(i+k-1)}, \dots, Y_{m(i+k-1)})
 \end{aligned}
 \tag{5}$$

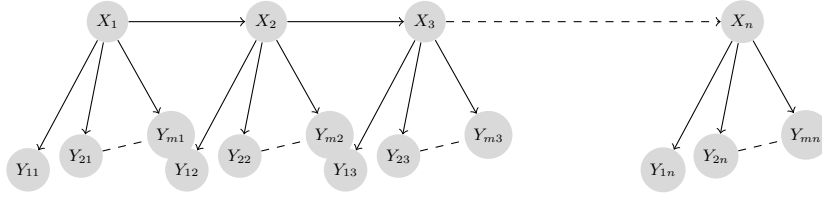


Fig. 2: Structure of Dynamic Bayesian Network(DBN)

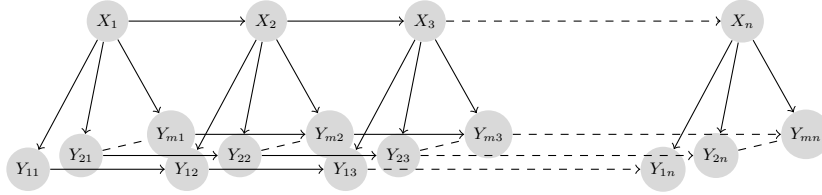


Fig. 3: Structure of Auto Regressive Dynamic Bayesian Network(AR-DBN)

After the sum product algorithm completes, we can use the marginal probability distribution to estimate the parameters in the network based on EM algorithm.

## 4 Formulation of Auto Regressive Dynamic Bayesian Network

As mentioned in Section 2, due to the ubiquitous auto regressive relationship in stock prices, we can add directed auto regressive edges between neighboring observed variables in the network. The resulting network is shown in Fig. 3. For each observed variable  $Y_{ki}$ , with the new assumption, it is not only conditioned on the latent variable, but also influenced directly by the previous observed variable.

For each observed variable, we have

$$Y_t = \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + \dots + \beta_k Y_{t-k} + \alpha U_t \quad (6)$$

Where  $k$  is the depth of the regression,  $Y_{t-k}$  to  $Y_{t-1}$  is the previous observed variables and  $U_t$  is directly emitted by the latent variable  $X_t$ . The coefficients satisfy  $\beta_1 + \beta_2 + \dots + \beta_k + \alpha = 1$ .

Denote  $Y'_t = Y_t - \beta_1 Y_{t-1} - \beta_2 Y_{t-2} - \dots - \beta_k Y_{t-k}$ , we have  $U_t = \alpha^{-1} Y'_t$ . From which we can estimate the parameters in AR-DBN, including conditional probability distribution  $P(X_i|X_{i-1})$ , the initial probability distribution  $P(X_1)$  and the emission probability  $P(Y_{1i}, Y_{2i}, \dots, Y_{ki}|X_i)$  which can be modeled as a multi variable Gaussian distribution  $N(\boldsymbol{\mu}, \boldsymbol{\sigma})$ .

Based on EM algorithm, we estimate the parameters with marginal probabilities from (4) and (5) by maximizing the first term of KL divergence

$$\begin{aligned} Q(\theta, \theta^{old}) &= \sum_X P(X|Y, \theta^{old}) \ln P(X, Y|\theta) \\ &= \sum_X P(X|\alpha^{-1}Y', \theta^{old}) \ln \left[ P(X_1|\pi) \prod_{n=2}^N P(X_n|X_{n-1}, A) \prod_{n=1}^N P(\alpha^{-1}Y'_n|X_n, \phi) \right] \\ &= \sum_X P(X|\alpha^{-1}Y', \theta^{old}) \ln P(X_1|\pi) + \sum_X P(X|\alpha^{-1}Y', \theta^{old}) \left( \sum_{n=2}^N \ln P(X_n|X_{n-1}, A) \right) \\ &\quad + \sum_X P(X|\alpha^{-1}Y', \theta^{old}) \left( \sum_{n=1}^N \ln P(\alpha^{-1}Y'_n|X_n, \phi) \right) \end{aligned}$$

Applying Lagrange method to fulfill the criteria  $\sum_{k=1}^K \pi_k = 1$  and  $\sum_{j=1}^K A_{ij} = 1$  for each  $i \in 1, \dots, K$ , where  $K$  is the number of states for the hidden variables, the parameters are derived by maximizing

$$R = Q(\theta, \theta^{old}) + \lambda_1 \left( \sum_{k=1}^K \pi_k - 1 \right) + \sum_{i=1}^K \lambda_{2i} \left( \sum_{j=1}^K A_{ij} - 1 \right) \quad (7)$$

After setting the first order partial derivatives of  $R$  with respect to each individual parameter to zero, the explicit expression of the parameters is derived as below

$$\pi_k = \frac{P(X_k|\alpha^{-1}Y')}{\sum_{j=1}^K P(X_j|\alpha^{-1}Y')} \quad (8)$$

$$A_{jk} = \frac{\sum_{n=1}^{N-1} P(X_n = j, X_{n+1} = k|\alpha^{-1}Y')}{\sum_{X_{n+1}=1}^K \sum_{n=1}^{N-1} P(X_n = j, X_{n+1}|\alpha^{-1}Y')} \quad (9)$$

$$\mu_i = \frac{\sum_{n=1}^N P(X_n|\alpha^{-1}Y') Y_{in}}{\sum_{n=1}^N P(X_n|\alpha^{-1}Y')} \quad (10)$$

$$\Sigma_i = \frac{\sum_{n=1}^N P(X_n|\alpha^{-1}Y')(Y_{in} - \mu_i)(Y_{in} - \mu_i)^T}{\sum_{n=1}^N P(X_n|\alpha^{-1}Y')} \quad (11)$$

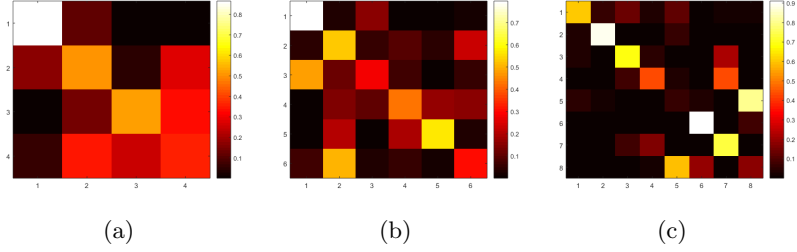


Fig. 4: The visualization of learned CPD. (a) 4 hidden states for each latent variable, (b) 6 hidden states for each latent variable, (c) 8 hidden states for each latent variable.

After the training phase completed, we infer the hidden states that form the highest likelihood path with max sum algorithm [8], similar to DBN. The inferred result for the stock market is shown in the next section.

## 5 Application in Stock Market

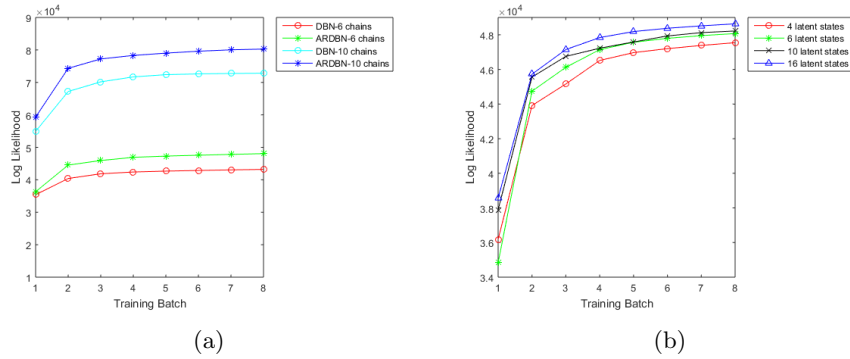


Fig. 5: (a) The comparison of log likelihood between DBN and AR-DBN with different number of observed chains. (b) The comparison of log likelihood between different number of latent states in AR-DBN.

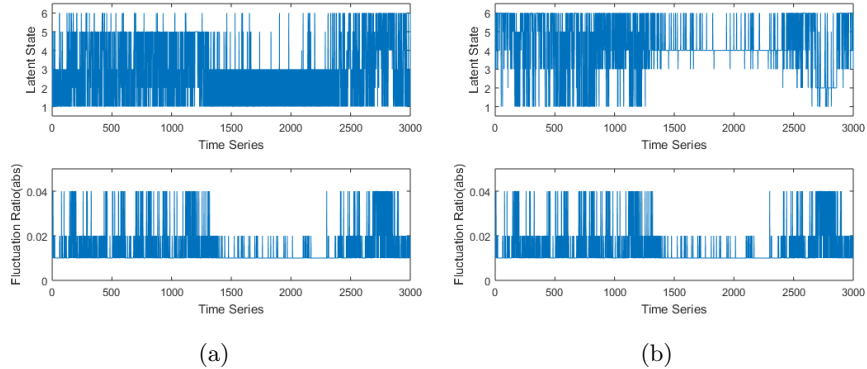


Fig. 6: Trend comparison of inferred latent state with *S&P* 500 fluctuation ratio. (a) Inference with 2 observed individual stock chains. (b) Inference with 6 observed individual stock chains.

We use the historical *S&P* 500 stock price dataset provided by Quantquote [16], covering the period from Jan. 02, 1998 to Aug. 09, 2013. The individual stock price temporal fluctuation forms each observed chain and the hidden states are inferred from the multiple observed chains, as shown in Fig. 3. We randomly pick  $k$  individual stocks out of the dataset, where  $k$  varies in  $[2,12]$ , then we infer the hidden states and parameters from the data. The trained CPD is visualized and shown in Fig. 4. From which we can see for the inferring task of 6 individual stocks ( $k = 6$ ), 8 hidden state is an overkill with first 3 hidden state actually not functioning in transition, while 4 hidden states are not enough to represent all different positions of the market. 6 hidden states is the optimized choice for the model.

The AR-DBN outperforms DBN both in the ultimate likelihood achieved and also in the convergence speed, as shown in Fig. 5(a). It reveals that AR-DBN is more suitable and efficient to apply in stock market. The likelihood is also positive correlated with the number of latent states as shown in Fig. 5(b).

The inferred latent states with max sum algorithm is shown in Fig. 6. The latent states plotted in temporal order form the path that produces highest likelihood in the whole network. The absolute value of increase/decrease ratio in *S&P* 500 index is discretized into three intervals  $0 < ratio < 1\%$ ,  $1\% < ratio < 2\%$  and  $2\% < ratio$ , then we compare it with the evolving trend of hidden states in Fig. 6. It can be seen that with more observable chains included, the corresponding relationship between the latent states and the market index is more obvious and accurate. With 6 observed individual stock chains, the hidden states in the model is capable to precisely capture the fluctuation of market index. Take Fig. 6(b) for example, state 1 and state 2 capture the characteristics of the fast changing market, while the other states reflect the time when the market is smooth. The hidden states have a direct reflection of the individual stock prices if we



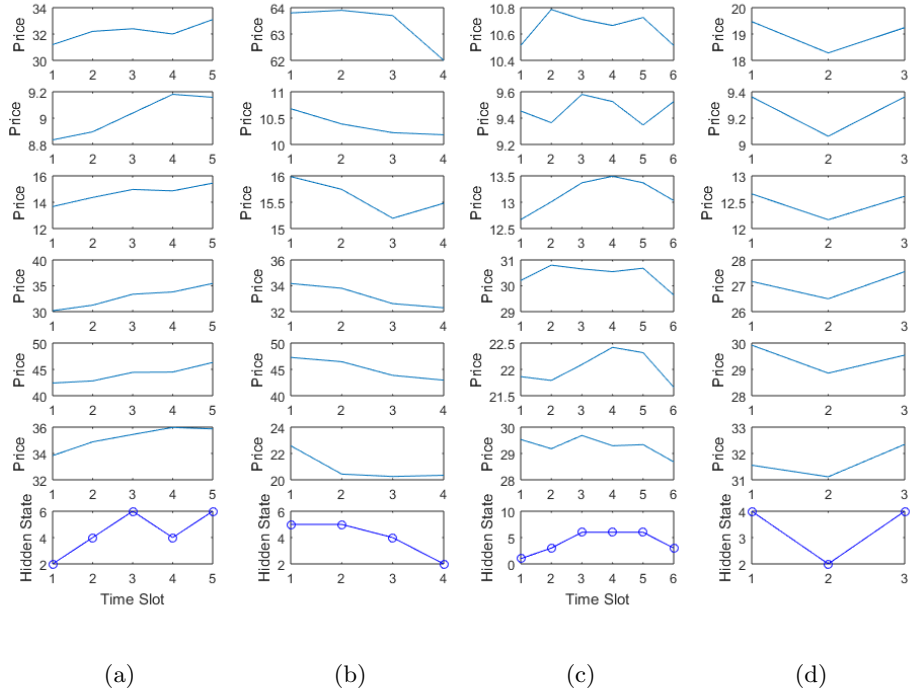


Fig. 7: The micro view of correlation in stock price trends and hidden states. (a) The ascending trend. (b) The descending trend. (c) and (d) The "V" reversal.

use absolute value of the price as the observed variable. Important trends in the price movement such as ascending trend, descending trend and "V" reversal are reflected in hidden states as shown in Fig. 7.

The AR-DBN stock market model we generated is not only useful to unveil market rules contained in historical stock price, it can also support investment decision making based on probabilistic prediction of the near future. In this application, we predict price movement direction(upward/downward) of the 6 observed stocks for the first day following the end of chain based on the parameters we learned. We multiply the latent states inferred from the max sum algorithm with the conditional probability distribution  $P(X_{t+1}|X_t)$  and turns the future stock price  $Y_{t+1}$  into a Gaussian mixture distribution conditioned on the probability distribution of future latent states  $X_{t+1}$ . The prediction result together with the real fluctuation for each of the stocks is shown in Fig. 8. For 5 out of the 6 stocks, the centroid of the probability distribution are of the same direction(upward/downward) with the real price trend, which shows our model can help with the prediction of fluctuation direction.

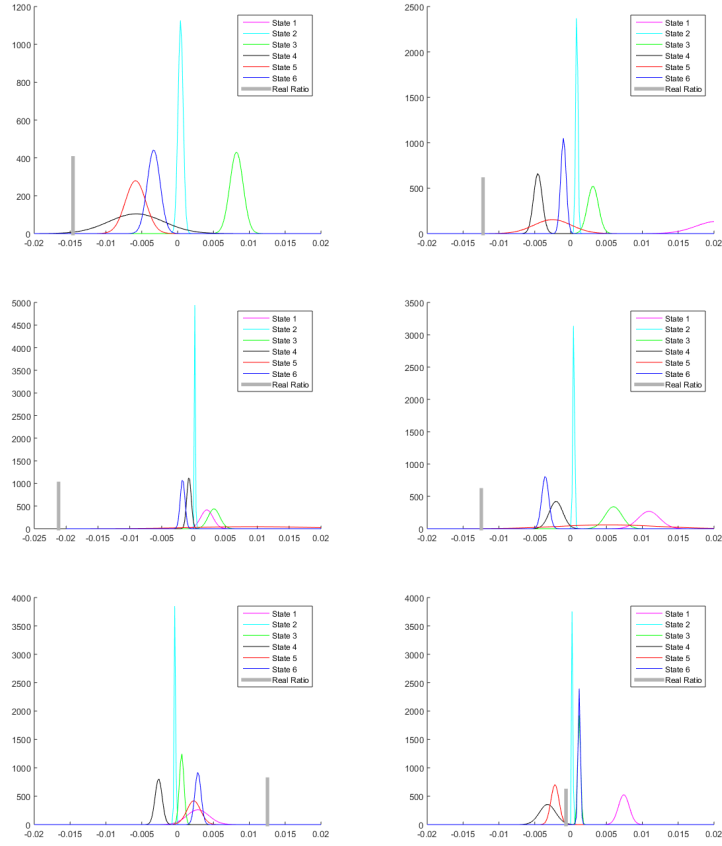


Fig. 8: The Gaussian mixture distribution of the predicted fluctuation and the real price fluctuation for each stock chain, from top left to bottom right corresponds to observed stock price chain 1 to 6.

## 6 Conclusion and Future Work

We derived Auto Regressive Dynamic Bayesian Network(AR-DBN) by adding auto regression among the adjacent observed variables in Dynamic Bayesian Network(DBN). We presented a new approach to model the stock market based on the proposed AR-DBN and comprehensive inference tasks were implemented with the model. The results showed the latent variables in the model accurately inferred the market index fluctuation and the stock price trends.

In this paper, we mainly focused on the inference part of the model. There is more work to be done with quantitative prediction of future price fluctuations and also the application of AR-DBN to other temporal analysis domains. We leave this for future work.

## References

1. Ai, B., Fan, Z., Gao, R.X.: Occupancy estimation for smart buildings by an auto-regressive hidden markov model. In: 2014 American Control Conference. pp. 2234–2239 (June 2014)
2. Barber, C., Bockhorst, J., Roebber, P.: Auto-regressive hmm inference with incomplete data for short-horizon wind forecasting. In: Lafferty, J., Williams, C., Shawe-taylor, J., Zemel, R., Culotta, A. (eds.) *Advances in Neural Information Processing Systems 23*, pp. 136–144 (2010), [http://books.nips.cc/papers/files/nips23/NIPS2010\\_1284.pdf](http://books.nips.cc/papers/files/nips23/NIPS2010_1284.pdf)
3. Barra-Chicote, R., Fern, F., Lutfi, S., Lucas-Cuesta, J.M., Macias-guarasa, J., Montero, J.M., San-segundo, R., Pardo, J.M.: Acoustic emotion recognition using dynamic bayesian networks and multi space distributions. *Annual Conference of the International Speech Communication Association (1)*, 336–339 (2009)
4. Bishop, C.M.: *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA (2006)
5. Das, B.: Generating conditional probabilities for bayesian networks: Easing the knowledge acquisition problem. *CoRR cs.AI/0411034* (2004)
6. Delage, E., Lee, H., Ng, A.Y.: A dynamic bayesian network model for autonomous 3d reconstruction from a single indoor image. In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06). vol. 2, pp. 2418–2428 (2006)
7. Jangmin, O., Lee, J.W., Park, S.B., Zhang, B.T.: Stock Trading by Modelling Price Trend with Dynamic Bayesian Networks, pp. 794–799. Springer Berlin Heidelberg, Berlin, Heidelberg (2004)
8. Jordan, M.I., Ghahramani, Z., Jaakkola, T.S., Saul, L.K.: An introduction to variational methods for graphical models. *Mach. Learn.* 37(2), 183–233 (Nov 1999)
9. Kampa, K., Principe, J.C., Putthividhya, D., Rangarajan, A.: Data-driven tree-structured bayesian network for image segmentation. In: 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 2213–2216 (March 2012)
10. Li, Z., Yang, J., Tan, S.: Systematically discovering dependence structure of global stock markets using dynamic bayesian network. *Journal of Computational Information Systems* 9, 7215–7226 (2013)
11. Likforman-Sulem, L., Sigelle, M.: Recognition of degraded characters using dynamic bayesian networks. *Pattern Recogn.* 41(10), 3092–3103 (Oct 2008)
12. Marcek, D.: Stock price forecasting: Autoregressive modelling and fussy neural network. *Journal of Information, Control and Management Systems* 7, 139–148 (2000)
13. Marcek, D.: Some intelligent approaches to stock price modelling and forecasting. *Journal of Information, Control and Management Systems* 2, 1–6 (2004)
14. Mathew, O.O., Sola, A.F., Oladiran, B.H., Amos, A.A.: Prediction of stock price using autoregressive integrated moving average filter. *Global Journal of Science Frontier Research Mathematics and Decision Sciences* 13, 78–88 (2013)
15. Mihajlovic, V., Petkovic, M.: *Dynamic bayesian networks: A state of the art* (2001), dMW-project
16. Quantquote: S&P500 daily resolution dataset
17. Rett, J., Dias, J.: *Image Analysis and Recognition: Third International Conference, ICIAR 2006, Póvoa de Varzim, Portugal, September 18-20, 2006, Proceedings, Part II*, chap. *Gesture Recognition Using a Marionette Model and Dynamic Bayesian Networks (DBNs)*, pp. 69–80. Springer Berlin Heidelberg, Berlin, Heidelberg (2006)

18. Shannon, M., Byrne, W.: A formulation of the autoregressive hmm for speech synthesis. Tech. rep., Cambridge University Engineering Department (08 2009)
19. Sipos, I.R., Levendovszky, J.: Optimizing Sparse Mean Reverting Portfolios with AR-HMMs in the Presence of Secondary Effects. *Periodica Polytechnica Electrical Engineering and Computer Science* 59(1), 1–8 (2015)
20. Suk, H.I., Sin, B.K., Lee, S.W.: Hand gesture recognition based on dynamic bayesian network framework. *Pattern Recogn.* 43(9), 3059–3072 (Sep 2010)
21. Tayal, A.: Regime Switching and Technical Trading with Dynamic Bayesian Networks in High-Frequency Stock Markets. Ph.D. thesis, University of Waterloo (2009)
22. Zweig, G., Russell, S.: Speech recognition with dynamic bayesian networks. In: *AAAI-98 Proceedings*. American Association for Artificial Intelligence (1998)

