Report of realization of Hidden-Markov-Model Prediction

Yike Wang

We realized stock price prediction in two ways based on the application of Hidden-Markov Models (HMM). The core idea of HMM is to transfer the problem of unpredictable prediction into prediction with observable and predictable features that are highly correlated with the target variables. In this context, we use observable sequences such as the momentum factors, mood factors, risk factors to predict the unobservable and elusive hidden states: the future trend of stocks and indexes.

Only three tools are needed in the iteration of HMM learning. The three tools are Transfer Matrix (Mat A), Emission Matrix (Mat B) and Prior Distribution (Array π). While the existence of casual link between the observation sequences and the hidden states is the prerequisite of the feasibility of the method, Mat B here is the one that relates the occurring probabilities of corresponding observation sequences to each contemporaneous hidden state. Mat A gives expression to observation sequences' predictability by containing the probabilities of each hidden state at current time is transferred from each hidden state at the last time. Prior probability offered by Array π is also of necessity since it gives clue to the starting point of each and every predicted hidden state path.

It is not hard to infer the significant part the selection of observation sequence plays in the whole prediction. The degree of correlation or say the casual link determines the smoothness of transferring the tricky direct prediction into handy indirect prediction. Therefore, in terms of observation sequence, we used factors that are all highly intertwined with prices and reflect the dynamic change or comparisons with the extremes. The features we used can be categorized into the mood factors, momentum factors and the risk factors. Despite of the differences among names, they are all essentially the same as various calculations based on prices. The feature pool we consider for the selection of the observation sequences are displayed in Table 1, while the final features filtered out by the Scoring Method are displayed in Table 2.

Table 1 Feature Pool for the Selection of Observation Sequences

Category	Feature Factors	Calculation
Mood	VROC12	12-Day Volume Change Rate
	TVMA6	6-Day Moving Average of Turnover
	DAVOL5	$\frac{5 - \text{Day Average Turnover Rate}}{120 - \text{Day Average Turnover Rate}}$
	WVAD	$\sum_{t-5}^{t} \frac{\text{ClosePrice} - \text{OpenPrice}}{\text{HighPrice} - \text{LowPrice}} \times \text{Volume}$
	VSTD20	Standard Deviation of 20-Day Volume
	AR	$\sum_{t-(N-1)}^{t} \frac{\text{HighPrice} - \text{OpenPrice}}{\text{OpenPrice} - \text{LowPrice}}$
	BR	$\sum_{t-(N-1)}^{t} \frac{\text{HighPrice} - \text{LastClosePrice}}{\text{LastClosePrice} - \text{LowPrice}}$
	ARBR	AR-BR
	VOL5	5-Day Average Turnover Rate
	turnover_volatility	Standard Deviation of 20-Day Turnover Rate

Momentum	BIASN	$\frac{\text{ClosePrice} - \text{N\_Day Average ClosePrice}}{\text{N\_Day Average ClosePrice}}$
	CR20	$\sum_{t-19}^{t} \frac{\text{HighPrice} - \text{LastMidPrice}}{\text{LastMidPrice} - \text{LowPrice}}$
	fifty_two_week_close_rank	Price Ranking among past 250 Trading Days (degrade)
	ROC12	$\frac{\text{ClosePrice}_t - \text{ClosePrice}_{t-12}}{\text{ClosePrice}_{t-12}}$
	single_day_VPT	$\frac{\text{ClosePrice} - \text{LastClosePrice}}{\text{LastClosePrice}} \times \text{Volume}$
Risk	VarianceN	N-Day Variance of Annualized Return
	SkewnessN	N-Day Skewness of Annualized Return
	KurtosisN	N-Day Kurtosis of Annualized Return
	Sharpe_ratioN	$\frac{\text{Return} - \text{RiskFreeReturn}}{\text{Standard Deviation of N} - \text{Day Return}}$

Table 2 Features Filtered Out by the Scoring Method

Category	Feature Factors
Mood	BR26
Momentum	BIAS60
Risk	Variance20
Others	Pe_ttm Close_SPX

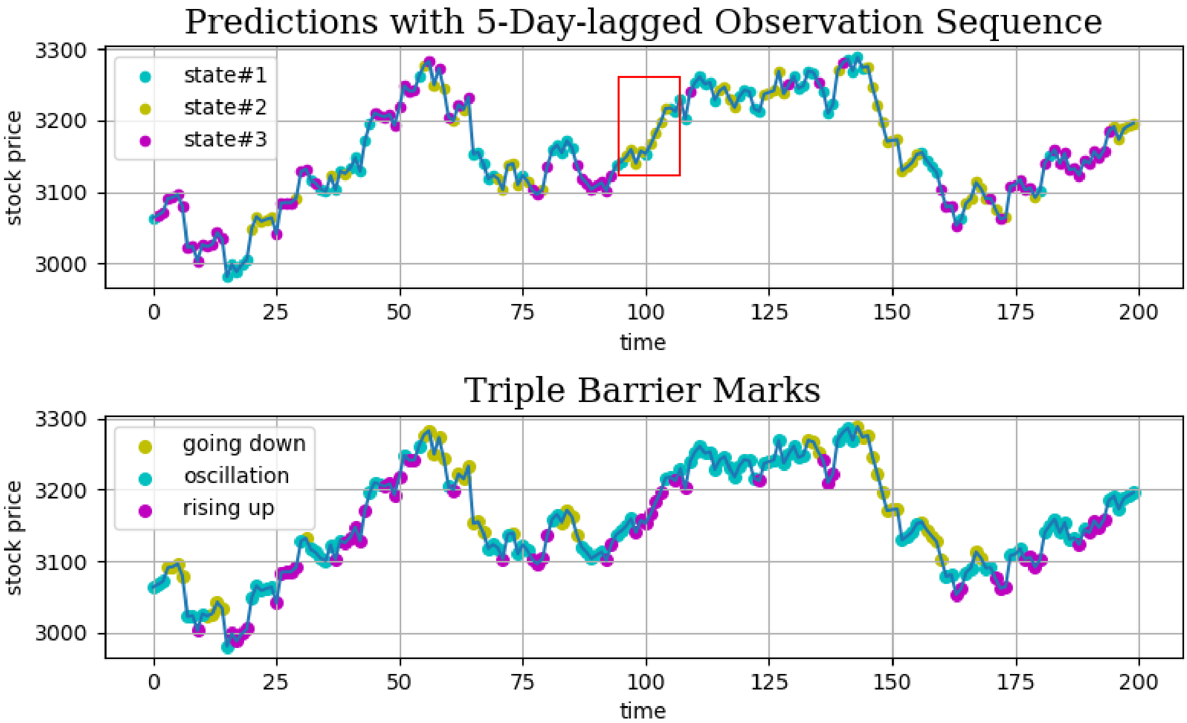
Readers shall mind that in the setting of financial stock market, Viterbi Algorithm (The algorithms that outputs the optimal hidden states path with the input of predictable contemporaneous observation sequences) needs to be used with consciousness, because the observation sequences are highly intertwined with contemporaneous stock prices, with which the usage would be of illegally forward looking. Therefore, in the prediction section, to consciously avoid Forward Looking Measurement, contemporaneous observation sequences are abandoned. We tackled the tricky problem by (I) Use contemporaneous observation sequences when the model is learning, but not during the prediction session. The learnt final transfer matrix will be the only item needed during the prediction session, making the rolling prediction a plain and simple Markov Chain Process; (II) Use lagged observation sequences as a substitution of predicted contemporaneous observation sequences as the input of Viterbi Algorithm.

(I) Results Sharing

The number of hidden states can be set according to personal preferences. The open source code we offer can support the hidden states to be any number ranging from 1 to 7. If the number is set to be 3, the unsupervised prediction results derived from HMM can be compared with the labels set according to Triple Barrier Labelling Method which labels stock price state to 1 when the price first touch the ceiling within horizontal time limit, -1 when first hit the floor and 0 when first hit the horizontal time limit. Since in the scenario above, the predictions and the labels obtained from the Triple Barrier Method is comparable, an accuracy rate can be calculated. Table 3 below is an accuracy result table of one try-out when we set the number of hidden states to be 3 and use only the filtered features.

Readers might be confused about how is results of predictions comparable while one is clustered supervision-free and the other labeled with deliberate supervision. Well, in order to provide the community we're lucky to have with intuitive results and higher predicting accuracy, our open source codes are capable of matching the unsupervised predicted states with the supervised labelled (-1,0,1) states before output the accuracy rate. The matching algorithm can be traced in the codes where I used dictionaries to depict the mapping of correspondence and calculate the degree of matching and choose the one mapping with the highest overall score for the output accuracy.

If we use all the features in Table 2, the best predicted results from rolling window predictions would be like what is shown below in Graph 1, where the purple dots stand for stock prices will go up, the blue dots stand for oscillation and the yellow dots stand for going down. The graph at top is the predicted result gained from unsupervised learning, the graph at bottom is the labels by Triple Barrier Method. We can see that the blue dots are quite well given in predictions, as it stands for oscillation itself. Moreover, the purple dots given from predictions also fit along with the real condition quite well. However, the result provided is quite too conservative, when we shift our eyesight into area framed by the red box in the graph at top, we can easily acknowledge that the place where "buy-in" signals should have been roaring is replaced by negative "sell-out" signals, losses was caused out of conservation. Since this is the best fitted part of the best predicted result, the roughness of the predictions with all the features in Table 2 is to be questioned. To refine simplification, we filtered out the most determinant factors with the Scoring Method and used the factors to get our final results.



Graph 1 Predicted Results with All Features

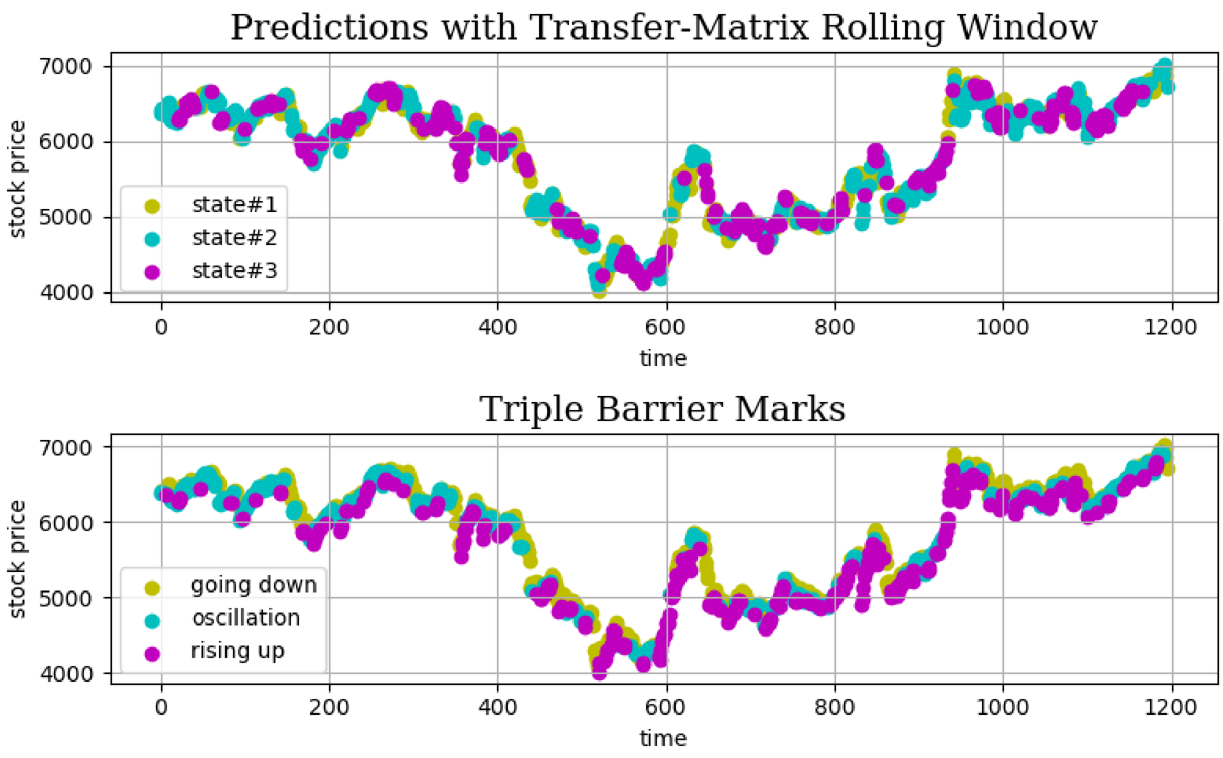
Accuracies are recoded neatly below in Table 3 when we predicted the stock price states with the simplified 5 factors. We came to the conclusion that: XGBoost is better than Gaussian-Mixed-Model when it comes to depicting the emission matrix. Readers shall mind that the code we provide here is the very basic set, where the initialization of parameter iterations is set randomly rather than by empirical prior experience. Therefore, any optimization done might bring tremendously benign changes to the predicting accuracy.

Table 3 Features Filtered Out by the Scoring Method

Predicting Method	**Emission Matrix**	**Accuracy (all features | filtered features)**			
		000001.SH	000016.SH	000905.SH	399001.SZ
Transfer Matrix	GMM	37.12%	36.37%	35.78%	36.75%
	XGBoost	37.50%	36.54%	36.50%	36.15%
Lagged Features	GMM	40.21%	37.72%	37.97%	36.63%
	XGBoost	44.31%	41.19%	37.16%	38.17%

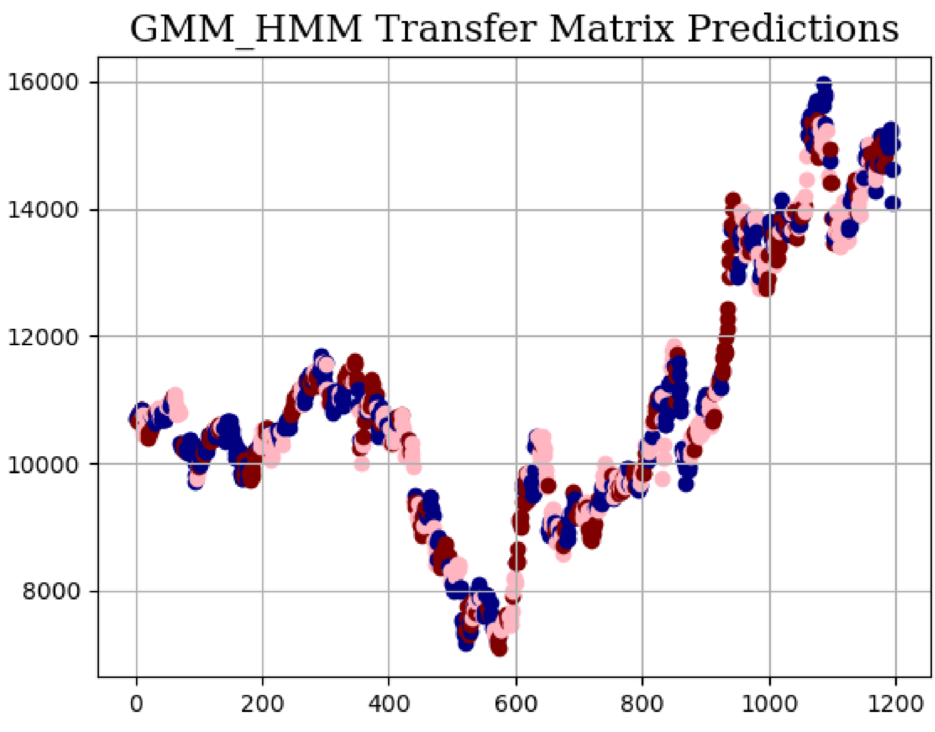
There are two ways to see the results. First, read the graph from an elbow's distance to check the overall consistency of the color between the graph at the top and at the bottom. This is to check for the matching degree of the predicted results given by unsupervised predictions with the labels given according to Triple Barrier Method. We depict the matching degree to be creditable and agreeable when the color tone is shared by predicted and real, for example Graph 2 is regarded as consistent, for the reasons as follow: when the stock price is trending upwards, both the real labels and the predicted ones are in a purple tone; During the first 400 days, when the real labelling at the bottom gives out a mixed blue tone, the blue sights are traceable as well in the predicted labels. The second way to see the results is to take a closer look at each obvious upslope and downslope of the stock index's price curve. It is okay to mix up oscillation states with the up and down states, but unforgivable to predict in the extremely opposite direction, to exemplify: predict the upward trend as downward or downward as upward. Readers need to mind that, all the results below are derived from 100-Day- Training-5-Day-Forecasting rolling window prediction, which means that no Forward Looking Method is stepped onto, and no future data is come across. Moreover, each and every predicted label at the top is only effective within a 5-day horizontal time span: one purple dot is conveying the message that the price is predicted to hit the ceiling before hitting the horizontal time limit or the floor. The ceiling is set to be +2% in terms of return, while the floor is set to be -1% after closer observation of the average rate of returns in 4 years and the concerns of the rule from Behavior Finance: The grief of losing can only be compensated with gaining at least 2 times of losses.

As we explained in the preceding context, we used two ways to predict the stock prices. The optimal results we got from two methods are both displayed below. Graph 2,3 show the results of the prediction with the last learnt Transfer Matrix where prediction is seen as a plain Markov Chain for simplification. Graph 4,5 display the results of the prediction with lagged features as substitutions of contemporaneous observation sequences, in the prerequisite that the fact "lagged past features are highly correlated with current spot's future states" is more than random daydream.



Graph 2 Predicted Results with Transfer Matrix Only and from Simplified Features

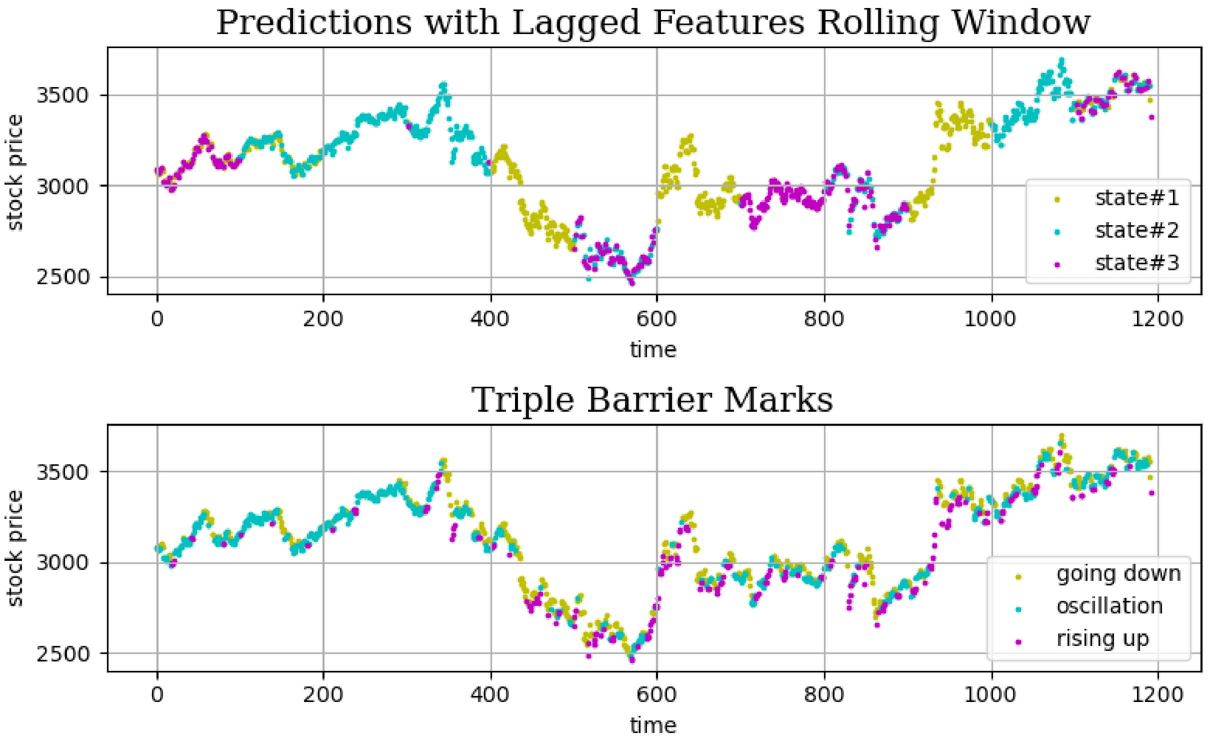
Graph 3 uses three different colors to depict the state, it is backed up by big data and large-scale tryout that the red dots stand for upward trending, the blue dots for downward trending and the pink ones for oscillation. It is significantly distinguishable from Graph 3.



Graph 3 Predicted Results with Transfer Matrix Only and from Simplified Features

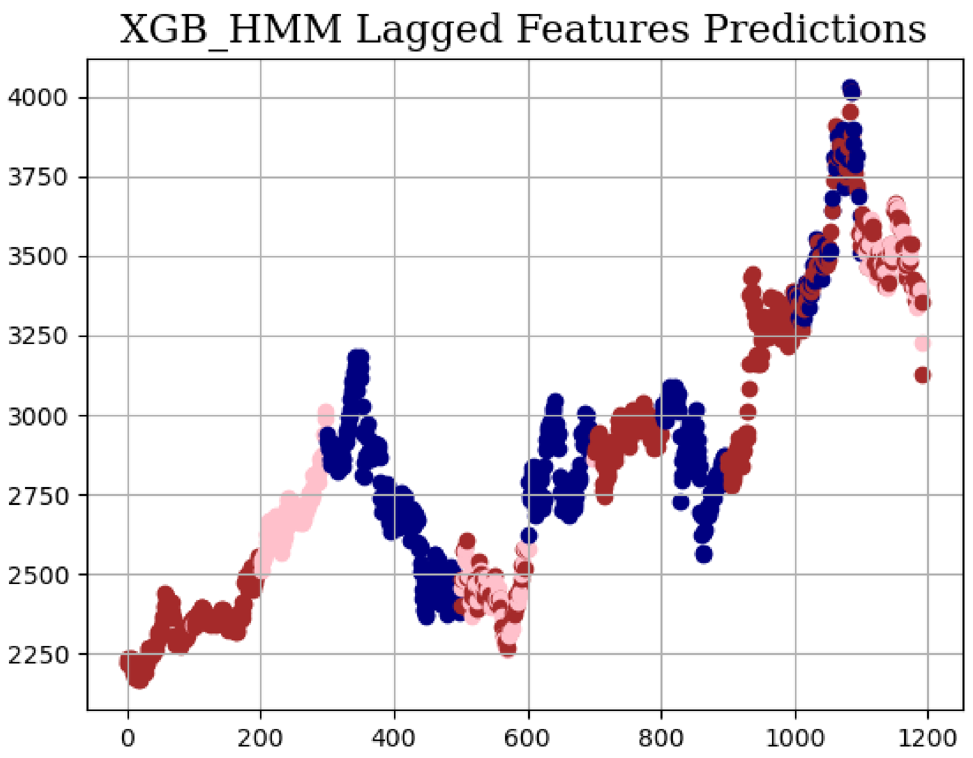
After adjusting the size of dots, readers can now have a clearer raw-eye judgment. We can see in Graph 4 that in the first 700 days, predictions are very accurate, giving out “wait and see” oscillation signals during 100-400 and “sell-out” signals during 400-500, “buy-in” signals both during 0-20 and 700-800. However, the 600-650 and 900-1000 parts are poorly predicted which could have caused regretful losses.

The colors below are obviously consistent when we read the graph from an elbow’s distance, however, it is also easy to acknowledge that the graph at the top is purer while the graph at the bottom as well as the result graph in Graph 2 with Transfer Matrix Method is more mixed in colors. We boldly infer that it is the clustering effect caused by Lagged Features Method. When we use the Lagged Feature Method during predictions, we obtain predictions in a different way than Multistep-Markov-Chain Predictions. In this scenario, we put in the entire forecasting window’s (default:5-Day) observation sequences and obtain the entire path with Dynamic Programming at once, that is to retrace the entire path with the maximum probability of overall occurrence from the end rather than get the next most highly likely forthcoming state from beginning to the end with Transfer Matrix Method. It might because that the Emission Matrix is more reluctant to the change of Observation Sequences compared to the sensitivity of the power of Transfer Matrix.



Graph 4 Predicted Results with Transfer Matrix Only and from Simplified Features

It is such a surprise to see the result of Graph 5, where the blue dots almost cover all the downward drops and the red dots cover all the upward rises.



Graph 5 Predicted Results with Transfer Matrix Only and from Simplified Features

(II) Handy Manuscripts

There are four code files in this GitHub branch, the codes are all Object-Oriented Programming in Python Language. Among the four code files, class_hmm_xgb_base.py and class_hmm_gmm_base.py are the two base files, Lagged_Feature_Prediction and Transfer_Matrix_Prediction are the two files that realize the rolling-window predictions and visualizations.

You can either choose to copy all the codes onto your python running panel or download all code files into local. Only need to mind that the four files needed to be in a folder and under the same path. For readers' convenience, I've packed the functions and features to be a whole, prediction can be easily made only use one simple command as exemplified in the two predicting files. Hope this is helpful in any extent.

(III) Acknowledgment

We would like to show gratitude to Guangzhou Shining Midas Investment Management Co., Ltd. For excellent support and resources. The whole experiment was based on one of the most heated topic and project belonging to Likelihood Lab in 2018. It is the past achievements done by the 2018 group and the face-to-face instructions provided by Maxwell Liu that exposed us to such advanced researches and guaranteed the smoothness of this whole open-source process. Without all those above, everything here would be nowhere to be found. Thanks a lot!

(IV) References

[1] Liu et al.,2021. Stock Market Trend Analysis Using Hidden Markov Model and Long Short Term Memory.

[2] HMM/GMM_hmm.py · YuHong-LDU/Python-AI - Gitee.com