Final Report of Data Mining

Multi-ways to Predicitor on Stocks

Chang Zhou
StudentID:5130309787
Class:F1303027
School of Electronic Information and Electrical Engineering
Shanghai Jiao Tong University
lemonbirdy@sjtu.edu.cn

2016/06/25

Abstract

Chapter 1

Introduction

Chapter 2

Stocks Predictior based on HMM

A Hidden Markov Model (HMM) is a finite state machine which has some fixed number of states. It provides a probabilistic graph model for series of observations. Hidden Markov models were introduced in the beginning of the 1970's as a tool in speech recognition. In recent years researchers proposed HMM as a classifier or predictor for speech signal recognition, DNA sequence analysis, handwritten characters recognition, natural language domains etc. HMM is a very powerful tool for various applications. Stocks data can be directly looked upon as a time series of observations which source from trading. Thus, it may become effective to build a predictor on stocks using HMM models. The key issue is that how to transfer the continous features like price, to discrete on with several states.

2.1 Brief Introduction to HMM

HMM is a graph model specilzed in time-sequence modeling. It holds assumption that the latent varible is only dependent on its previous latent variable while the observation varible is only dependent on its corresponding latent variable. The latent variables form a Markov chain. Thus the observation variables can related to all the others through latent ones. Meanwhile, it avoids the large computation that n-order Markov model brings. Here we list some of the advantages of HMM which are:

1. HMM has strong statisical foundation
2. It is able to handle new data robustly
3. Computationally efficient to develop and evaluate due to the existence of established training algorithms
4. It is able to predict similar patterns efficiently

2.2 HMM Prediction Model

2.2.1 Basic Notation

Hidden Markov Model is characterized by the following:

1. M: number of states in the model
2. $\{O\} = \{O_1, O_2, ..., O_T\}$: observation sequence
3. $A = \{a_{ij}\}$: state transition probabilities
4. $B = \{b_j(O_t)\}$: observation emission probability distrbution
5. $\pi = \{\pi_i\}$: initial state distribution

2.2.2 Data Preprocessing

In our model, the target is to predict whether the next day's closing price increases or decreases for a specific stock market share using a HMM model. 80% of past daily data were used for training while 20% of past daily data were used for testing. The observations data features are opening, high, low, and closing price's linear combinations. We have 12 stocks. In each stock we take

high-open	low-open	close-open

time sequence length as 20.

2.2.3 Continous HMM Model

Here is one of the key issues when we apply HMM model on stock prediction. The input is continuous. Therefore, we need to transfer it into variable with several states. The tool we use is Gaussian Mixture model. We apply GMM to the observation like the method we use in speech signal recognition.
After the GMM, the observations are transferred into states according to probability density function. Then we use it model the continous HMM.
We first give initial value to the transition matrix, emission probabilties and initial probabilities. After initialization, we trained the HMM using Baum Welch algorithm, a common EM method.
Now that we have got the trained HMM model. There comes another key issue that HMM don't provide directly classification or regression result. Instead, it provide loglikelihood probability for each testing sequence. We can do classification according to the value when it reach the maximum.

$$output = argmax(Likelihood)$$

where result can be either to increase or to decrease.

2.3 Experiment

To get the best result, we tried several combination of number of hidden state, s and number of mixture in GMM, m. The combination that has the best performance is that $(s, m) = (6, 8)$ The first table is about the output for close-open of next day. If it is positive, then it mean the stock would go up otherwise go down. Due to the space, we list some of the results.

Stock	stock1	stock2	stock3	stock4	stock5	stock6	stock7	stock8
Accuracy	55.28%	52.03%	57.63%	56.81%	58.93 %	52.19%	51.77%	52.26%

The second table is about the continous number of how much it go up or down. The measure method is RMSE.
The result are not so pleasant. Both discrete predicter or continous predicter

Stock	stock1	stock2	stock3	stock4	stock5	stock6	stock7	stock8
Accuracy	48.27%	36.89%	49.02%	53.08%	50.29 %	75.43%	57.12%	47.26%

did just a litter better than the random generators. The procedure let me know that train HMM to predict is not a simple way. They require so many prior knowledge. If we choose a bad initialization, then even after interations, the result is till not good.

2.4 Conclusion and Future Work

There are three aspects we need to improve:

1. The tutorial about GMM-HMM models are about one-input, so how to deal with multi-dimension input has become a problem? The GMM's assumption on the distribution of input observation is strong, maybe k-means would be better.

2. The HMM is more like a analysizer. We found that when we use HMM to do predictor, discrete output can be somehow easier, but continous one can be much more trivial.

3. The initialization and computation on iterations has become the difficulty of realization. Later we have found some paper has an introduction on the method of how to initialize. Those may have a good improvement on our methods.

Chapter 3

Reference

Hassan M R, Nath B. Stock market forecasting using hidden Markov model: a new approach[C]//5th International Conference on Intelligent Systems Design and Applications (ISDA'05). IEEE, 2005: 192-196.
http://www.cs.ubc.ca/ murphyk/Software/HMM/hmm_usage.html https://github.com/PhilipGeng/stockPred