1 引言

1.1 符号

符号	意义
T	length of the observation sequence
N	number of states in the model
M	number of observation symbols
A	state transition probabilities
B	emission probability matrix
π	initial state distribution
$O = \{O_0, O_1, .., O_{T-1})\}$	observation sequence
$S = \{S_0, S_1, ..., S_T T - 1\}$	state sequence
$V = \{V_0, V_1, ..., V_{M-1}\}$	set of possible observations
$\lambda = (A, B, \pi)$	HMM model
ϕ	score function

2 数据处理

2.1 特征工程

2.1.1 Y特征的构造

我们小组的目的是预测股票价格的涨跌，因此，我们需要构造一个能反应股票未来价格涨跌的Y特征。

我们使用the triple barrier method，这个方法由Marcos首次提出，是一种非常新颖的对特征打标签的方式，如图Fig.2.1.所示。

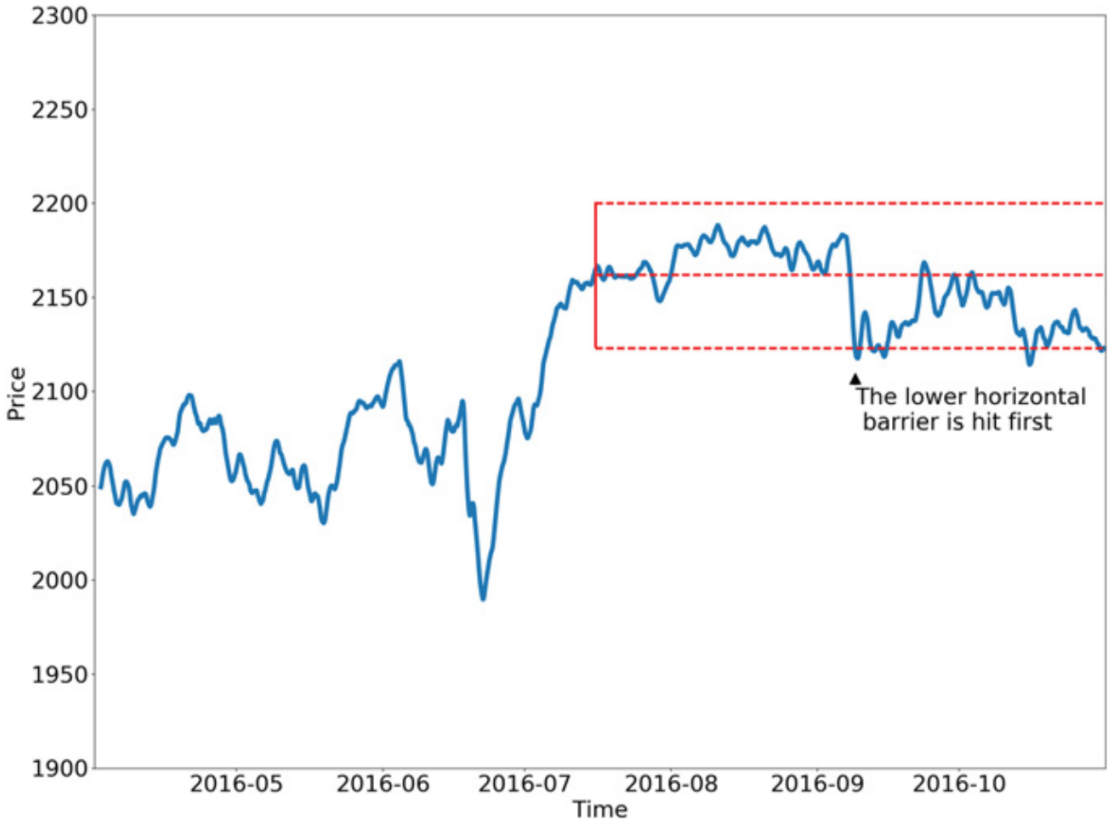


Fig.4.1 the triple barrier method结构

它设定了三个分界线，并根据路径触及的第一个屏障来做标记。

首先，我们设置了两个水平线和一个垂直线。两个横向分界线由价格涨幅定义范围。第三个界线是根据自采取该位置以来经过的柱数来定义的。

如果首先触及最上端的界线，我们将观测数据标记为1；如果首先触及垂直界线，标记为0,如果首先触摸最下端的界线，我们标记为-1。

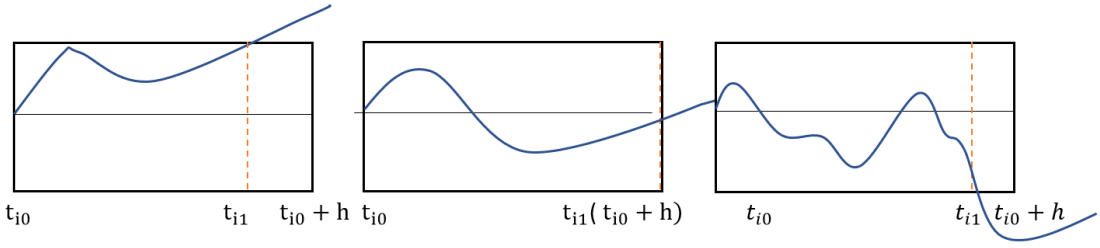


Fig.4.1 Label t_{i0} as 1,0 and -1

The triple barrier method 反应了股票未来价格的涨跌情况，符合我们小组的目的。

The triple barrier method的一个问题是路径依赖性。为了标记观测数据，我们必须考虑跨越$[t_{i0}, t_{i0}+h]$的整个路径，其中h定义垂直分割。我们将表示t_{i1}表示首次触碰分界线的时刻，与观察到的特征相关的返回是t_{i0}, t_{i1}。为了清楚起见，规定$t_{i1} \leq t_{i0}+h$，并且水平分界线不一定是对称的。

至此，Y特征的构造完成。

2.1.2 观测序列的构造

本小组将手头现有数据分为7种类型，分类如下：

类别	意义
行情类因子	包含昨日收盘价，今日开盘价，今日收盘价，交易量等指标
质量类因子	包括公司资产负债，周转，运营，盈利，成本费用等指标
收益与风险类因子	
价值类因子	包括公司市值，市盈率，市净率等指标
情绪类因子	包括换手率，动态买卖，成交量等指标
技术指标类因子	包括平均移动线，计算周期，动态移动等指标
动量类因子	包括平均移动，圆滑曲线，收益，未来趋势预测等指标
因子增长类	包括增长率等指标

构造特征方法如下：

我们小组构造一个分数函数ϕ，用以评估因子F的特征强度。

（1）固定一个因子F，将F的每日数据作为O，训练GMM-HMM，利用Viterbi算法生成最佳状态序列S。

（2）根据S，我们构造一个计数矩阵M,$M \in R^{N\times 3}$,其中M_{ij}表示状态i对应label j 的频数。$i \in \{0,1,...,N-1\}, j \in \{-1,0,1\}$

（3）根据M，我们构造计数比例矩阵MR,

$$MR_{ij} = \frac{M_{ij}}{\sum_j M_{ij}}$$

（4）状态i的准确率为

$$Acc_i = max_j\{MR_{ij}\}, i \in \{0,1,...,N-1\}, i \in \{0,1,...,N-1\}, j \in \{-1,0,1\}$$

（5）状态i的熵

$$H_i = -\sum_j MR_{ij} log(MR_{ij}), i \in \{0, 1, ..., N-1\}, i \in \{0, 1, ..., N-1\}, j \in \{-1, 0, 1\}$$

（6）状态i的权重

$$w_i = \frac{\sum_i M_{ij}}{\sum_i \sum_j M_{ij}}$$

（7）计算分数函数

$$score = \phi(Acc, H, w) = \sum_i (Acc_i \times \frac{1}{1 + H_i} \times w_i)$$

（8）分数越高，则认为因子F的特征强度越高。

选出的特征如下表。

类别	意义
行情类因子	一日对数收益差,五日对数收益差 当日对数高低价差,当日成交量 对数融资余额差
质量类因子	
收益与风险类因子	
价值类因子	
情绪类因子	
技术指标类因子	
动量类因子	
因子增长类	

3 GMM-HMM模型

3.1 引言

高斯混合——隐马尔可夫模型(GMM-HMM)被广泛应用于语音识别领域，并取得巨大的成功。

受此启发，我们小组认为，股票价格涨跌预测的任务可以总结为：给定已知的股票特征观测序列，找到概率最大的隐藏状态序列。然后利用LSTM找出隐藏状态序列与股票价格涨跌的关系。

得到的股票特征观测序列$O = O_1 O_2 ... O_T$,其中O_t代表在时间t上得到的股票特征。

一个隐藏状态序列为$S = S_1S_2...S_T$，其中S_t代表在时间t上的隐藏状态。

因此我们的任务可以总结为求使概率$P\{S|O\}$最大的隐藏状态序列s，即

$$s = argmax_S P\{S|O\}$$

本节将会介绍基于GMM-HMM模型的股票价格涨跌预测，阐述每个算法的核心以及在金融数据中的实证操作。

3.2 基本假设

（1）我们小组假设隐藏的状态符合齐次马尔可夫性假设，即假设隐藏的马尔可夫链的任意时刻t的状态只依赖于上一个时刻t-1的状态，与其他时刻的状态无关。

$$P\{S_t|S_{t-1}, O_{t-1}, ..., S_1, O_1\} = P\{S_t|O_{t-1}\}$$

（2）假设隐藏状态与股票特征观测序列符合观测独立性假设，即假设任意时刻的观测只依赖该时刻的马尔可夫链的状态，与其他观测及状态无关。

$$P\{O_t|S_T, O_T, S_{T-1}, O_{T-1}, ..., S_{t+1}, O_{t+1}, S_t, O_t, S_{t-1}, O_{t-1}, ..., S_1, O_1\} = P\{O_t|S_t\}$$

（3）混合高斯模型的方差是对角矩阵。

3.3 GMM-HMM模型

A GMM-HMM is a statistical model that describes two dependent random processes, an observable process, and a hidden Markov Process. The observation sequence is assumed to be generated by each hidden state according to a Gaussian mixture distribution.[2]

隐马尔可夫模型由初始概率分布向量π、状态转移概率分布矩阵A以及发射概率分布矩阵B确定。π和A决定状态序列，B决定观测序列。

GMM-HMM模型中，参数B是一个观测概率密度函数，它可以近似的表示为多个混合高斯分布的组合。通过确定每一个高斯概率密度函数权重w_{jk}，均值和协方差矩阵，可以将连续概率密度函数表示为：

$$b_j(V_t) = \sum_{k=1}^{M} w_{jk} b_{jk}(V_t), j = 1, ..., N, 0 \leq t \leq M-1$$

其中w_{jk}表示各成分的权重,

GMM-HMM模型结构如图Fig.4.1.所示。

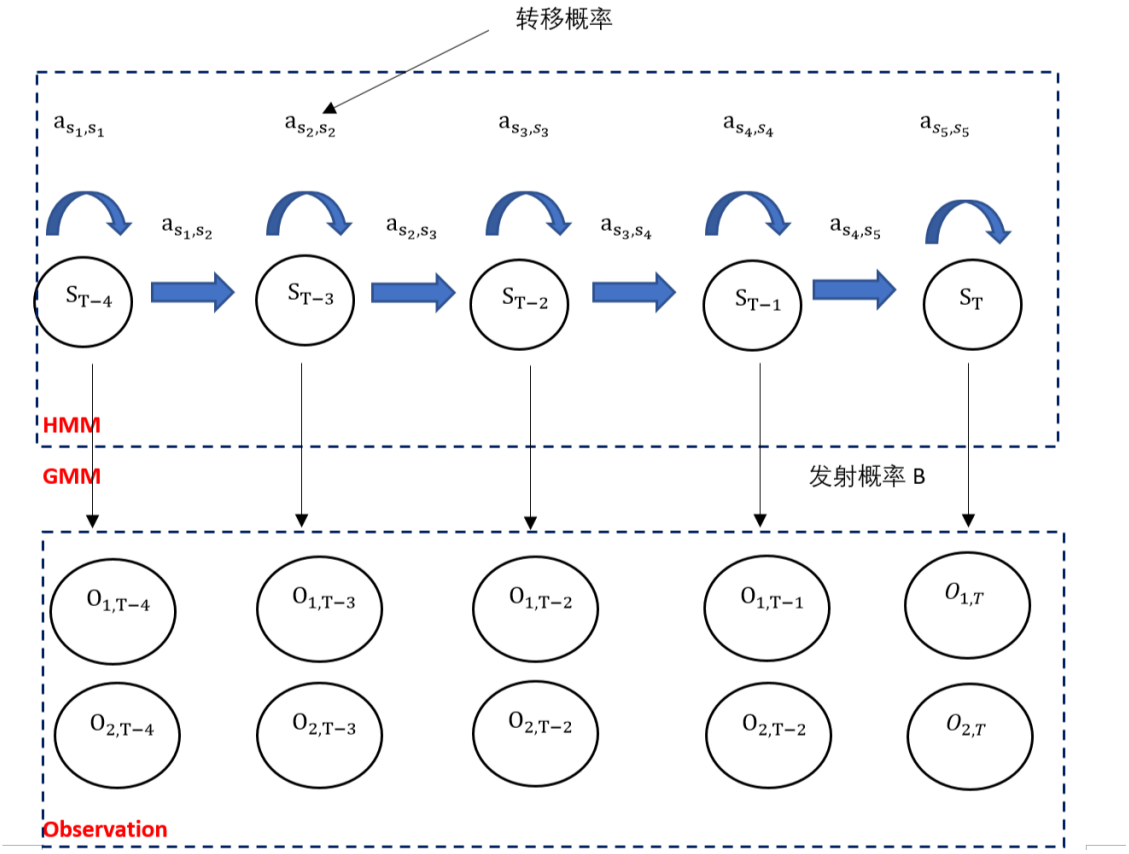


Fig.4.1 GMM-HMM模型结构

3.4 GMM-HMM模型训练算法

（1）确定隐状态个数。即确定N.

（2）给定观测序列O, 使用Baum-Welh算法估计模型参数$\lambda = (A, B, \pi)$。记该模型为gmm-hmm。

（3）在gmm-hmm模型中，使用Viterbi算法估计概率$P\{S_t = i\}, i = \{1, 2, .., N\}, t = \{0, 1, .., T-1\}$.

3.5 实验结果与分析

为了检验我们模型的效果，我们选取其中一只股票进行模型训练，确定模型的各种参数。

我们小组选取的股票为丰原药业，选取2007-01-04至2013-12-17的数据作为训练集。

我们选取隐状态数目N=3，选用行情数据作为特征，再把每一时刻预测到的状态与其对应的价格放在一起做可视化分析。

GMM-HMM训练集结果如下图，

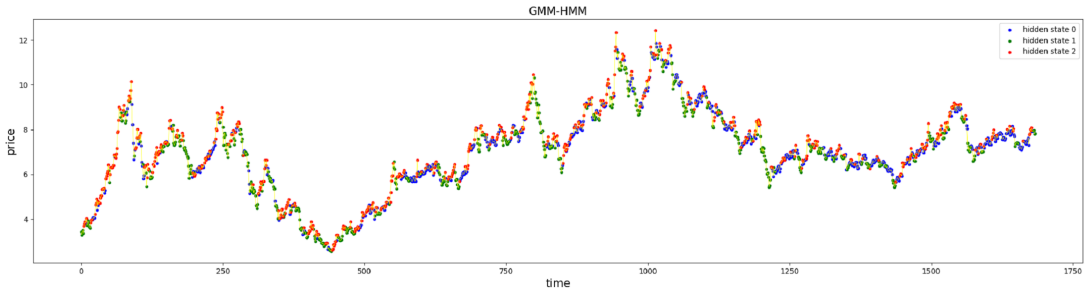


Fig.3.1 GMM-HMM训练集结果

从图中可以看出，橙色代表上升状态，绿色代表下降状态，蓝色表示震荡，而且直观来看，此GMM-HMM混合模型在训练集上的效果很好，记此模型为$gmm - hmm_0$.

将股票红太阳（000525.XSHE）2014-12-01至2018-05-21的数据作为训练集，运行模型为$gmm - hmm_0$.

GMM-HMM测试集结果如下图，

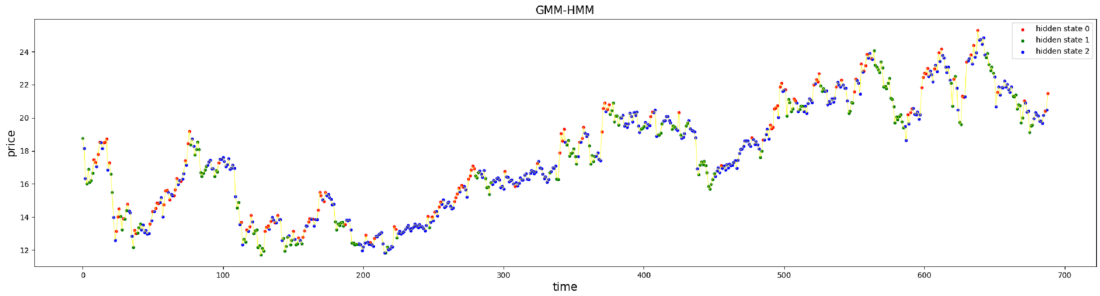


Fig.3.2 GMM-HMM测试集结果

3.6 模型优缺点分析

one major reason for this success is the use of the generative model of HMMs for representing the (piecewise stationary) dynamic speech pattern and the use of EM algorithm for training GMM.[2]

Despite great success of GMM-HMMs in speech modeling and recognition, their weakness, such as the conditional independence （没有利用上下文信息）and 不能学习深层非线性特征变换have been well known.[2]

EM算法对初始值敏感，不能保证找到全局最优点。

待补充。。。。。

4 XGB-HMM模型

4.1 引言

In speech recognition, the DNN-HMM hybrid system takes advantage of DNN's strong representation learning power and HMM's sequential modeling ability, and outperforms conventional Gaussian mixture model(GMM)-HMM systems significantly on many large vocabulary continuous speech recognition tasks.[1]

The DNN-HMM hybrid system is illustrated in Fig. 5.1.

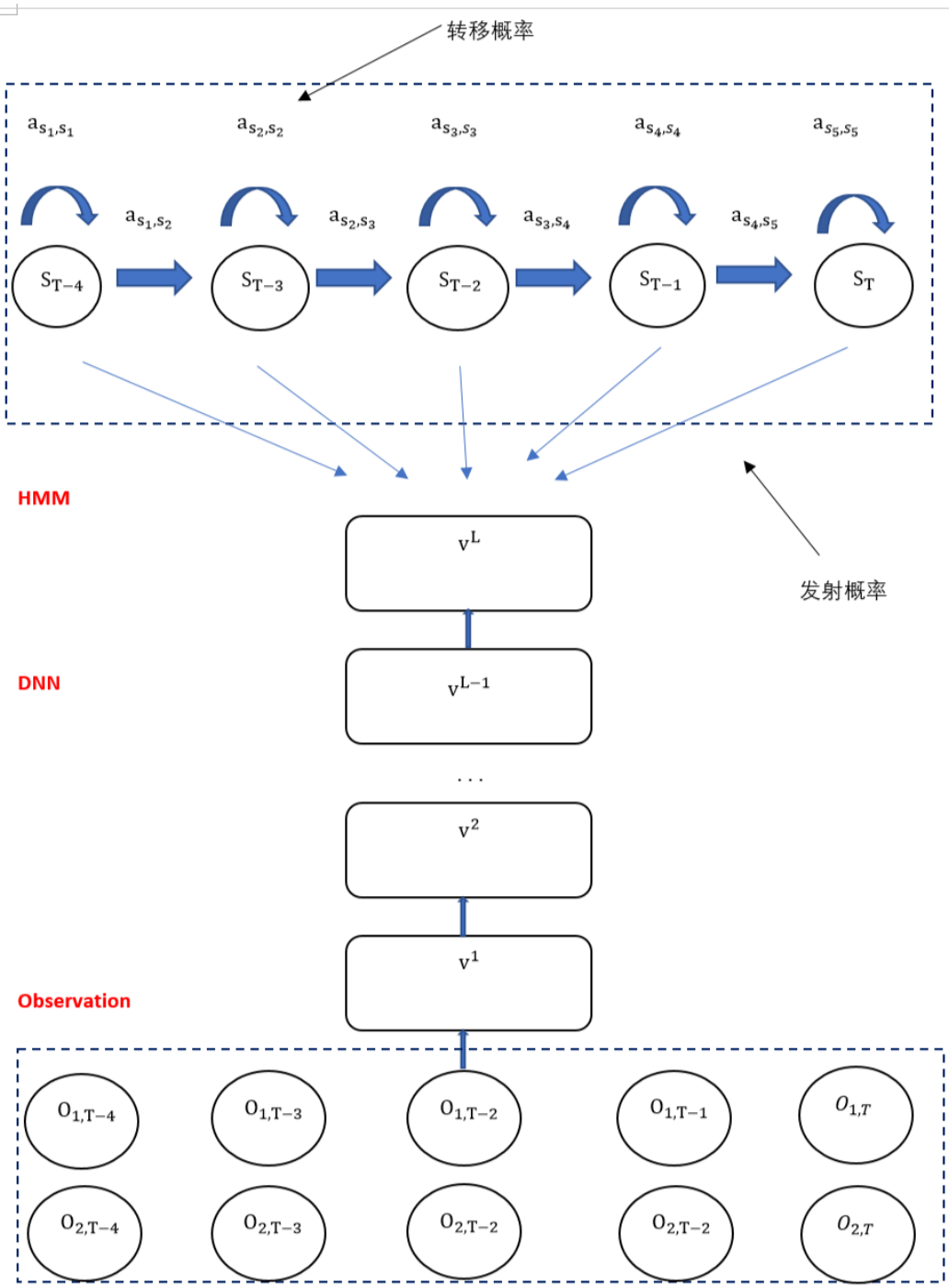

转移概率
a_{s_1,s_1}
a_{s_2,s_2}
a_{s_3,s_3}
a_{s_4,s_4}
a_{s_5,s_5}
a_{s_1,s_2}
a_{s_2,s_3}
a_{s_3,s_4}
a_{s_4,s_5}
S_{T-4}
S_{T-3}
S_{T-2}
S_{T-1}
S_T
HMM
v^L
发射概率
DNN
v^{L-1}
...
v^2
v^1
Observation
$O_{1,T-4}$
$O_{1,T-3}$
$O_{1,T-2}$
$O_{1,T-1}$
$O_{1,T}$
$O_{2,T-4}$
$O_{2,T-3}$
$O_{2,T-2}$
$O_{2,T-2}$
$O_{2,T}$

Fig.4.1 DNN-HMM模型结构

图中，所有的隐马尔科夫模型的状态对应同一个深度神经网络。

In the DNN-HMM, a single DNN is trained to estimate the conditional state posterior probability $P\{S_t|O_t\}$ for all states $S_t \in \{1, 2, 3\}$.This is different from the GMMs in which a different GMM is used to model each different state. In addition, the input to the DNN is typically not a single day of observation but a window of ω days of input features O_T to exploit information in the neighboring days.

$$O_T = \begin{pmatrix} O_{0,T-\omega}.....O_{0,T-1} \\ O_{1,T-\omega}.....O_{1,T-1} \\ ..... \\ O_{M-1,T-\omega}.....O_{M-1,T-1} \end{pmatrix}$$

In the light of this, we introduce a XGB-HMM model which exploits 可扩展的端到端树推进系统XGBoost instead of Gaussian mixture model (GMM) in estimating the emission probabiltiies $P\{O_t|S_t\}$.

在本节中，我们将会描述基于XGBoost的模型结构及其迭代训练算法。

4.2 XGB用于估计发射概率

4.2.1 XGB-HMM模型结构

在实践中使用的机器学习方法中，提升树增强被认为是统计学习中性能最好的方法之一。由于树的线性组合可以很好的拟合训练数据，即使数据中的输入输出之间的关系很复杂也是如此，所以树模型是一个高功能的学习方法。而提升树是指提升方法实际采用加法模型（即基函数的线性组合）与前向分布算法，以决策树为基函数的提升方法[3]。它是一种高效且广泛使用的机器学习方法。

XGB介绍。

4.2.2 XGB-HMM模型训练算法

初始化：

(1) 首先训练一个GMM-HMM模型,设训练完成的系统为gmm-hmm。得到$\lambda = (A, B, \pi)$.

(2) 利用gmm-hmm模型,使用向前向后算法求出$\alpha_t(i), \beta_t(i)$，继而求出$\gamma_t(i) = P\{S_t = i|O, \lambda\}$ 和$\gamma_t(i, j) = P\{x_t = q_i, x_{t+1} = q_j|O, \lambda\}$

For t=1,2,..,T-1 and i = 0,1,..,N-1

$$\alpha_t(i) = P\{O_0, O_1, ..., O_t, x_t = q_i|\lambda\}$$

$$\beta_t(i) = P\{O_{t+1}, O_{t+2}, ..., O_{T-1}|x_t = q_i, \lambda\}$$

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{P\{O|\lambda\}}$$

$$\gamma_t(i) == \frac{a_t(i)a_{ij}b_j(O_{t+1})\beta_{t+1}(j)}{P\{O|\lambda\}}$$

更新：

(3)根据

$$a_{ij} = \frac{\sum_{t=0}^{T-2}\gamma_t(i,j)}{\sum_{t=0}^{T-2}\gamma_t(i)}$$

更新转移概率A。

(4) 将O作为输入，$\gamma_t(i) = P\{S_t = i|O, \lambda\}$作为标注，训练XGB模型，记为$XGB_0$.

(5) 利用XGB_0估计$\hat{\gamma_t(i)}$, 记为$\gamma_t(i)_{new}$，然后利用

$$b_j(k) = \frac{\sum_{t\in\{0,1,..,T-1\}\&O_t=k}\gamma_t(j)_{new}}{\gamma_t(j)_{new}}$$

更新发射概率B。

至此，$\lambda = (A, B, \pi)$更新完毕。

(6)如果$P\{O|\lambda\}$上升，返回步骤2，继续更新λ；否则，结束训练。

The training algorithm can be summarized as follows.

1.Initialize,$\lambda = (A, B, \pi)$.

2.Compute $\alpha_t(i), \beta_t(i), \gamma_t(i), \gamma_t(i, j)$.

3. Train XGB.

4.Re-estimate the model $\lambda = (A, B, \pi)$.

5.If $P\{O|\lambda\}$ increases, goto 2.

Of course, it might be desirable to stop if $P\{O|\lambda\}$ does not increase by at least some predetermined threshold and/or to set a maximum number of iterations.

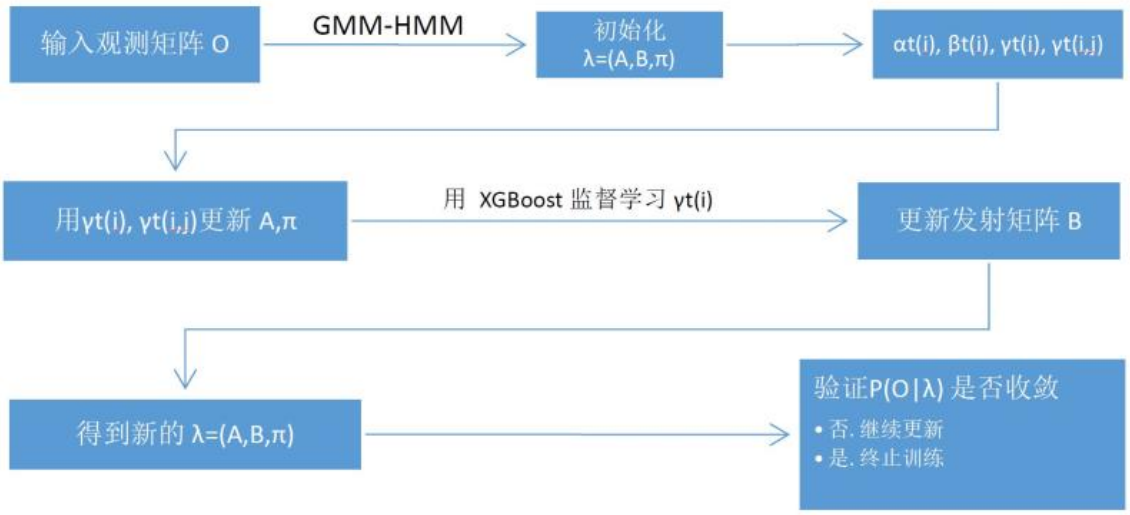


Fig.4.2 XGB-HMM算法示意图

4.3 实验结果与分析

为了检验我们模型的效果，我们选取其中一只股票进行模型训练，确定模型的各种参数。

我们小组选取的股票为丰原药业，选取2007-01-04至2013-12-17的数据作为训练集。

我们选取隐状态数目N=3，选用行情数据作为特征，再把每一时刻预测到的状态与其对应的价格放在一起做可视化分析。

XGB-HMM训练集结果如下图，

Fig.4.6 XGB-HMM训练集结果

从图中可以看出，橙色代表上升状态，绿色代表下降状态，蓝色表示震荡，而且直观来看，此GMM-HMM混合模型在训练集上的效果很好，记此模型为$xgb - hmm_0$.

将股票红太阳（000525.XSHE）2014-12-01至2018-05-21的数据作为训练集，运行模型

为$xgb - hmm_0$.

XGB-HMM测试集结果如下图，

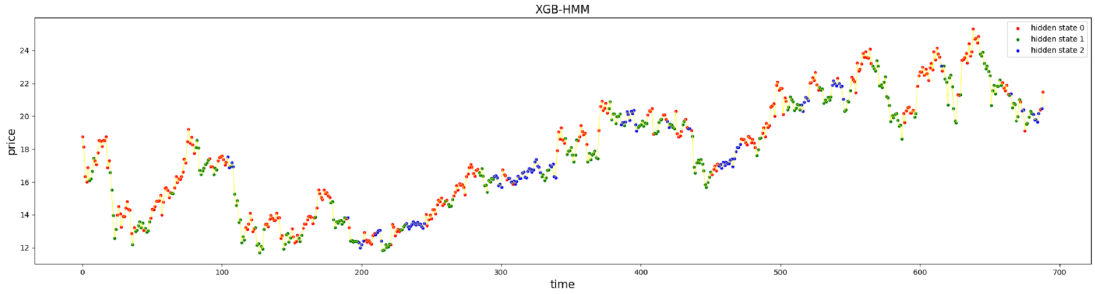


Fig.4.6 XGB-HMM测试集结果

XGB-HMM算法的迭代图如下。

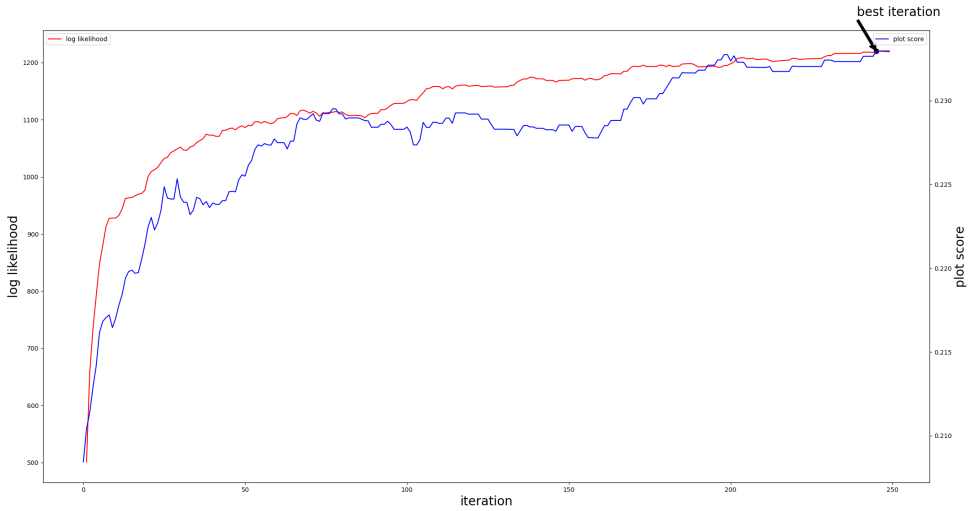


Fig.4.6 迭代图

从图中可以看出，随着迭代次数的增加，模型log-likelihood随之增加，当迭代次数达到约250次后，log-likelihood趋于平稳。同时，随着迭代次数的增加，plot score的分数也在增加。

下图为GMM-HMM和XGB-HMM测试结果的对比图。

可以看出，运用XGB-HMM混合模型的结果更好，使得三个隐状态的区分度更高，意味着XBG在拟合观测数据方面可能有更好的效果。

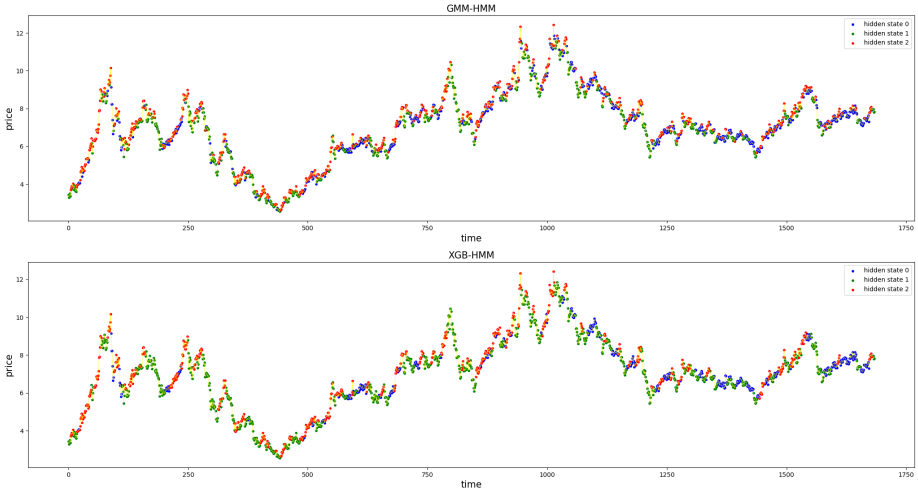


Fig.4.7 GMM-HMM与XGB-HMM训练集对比图

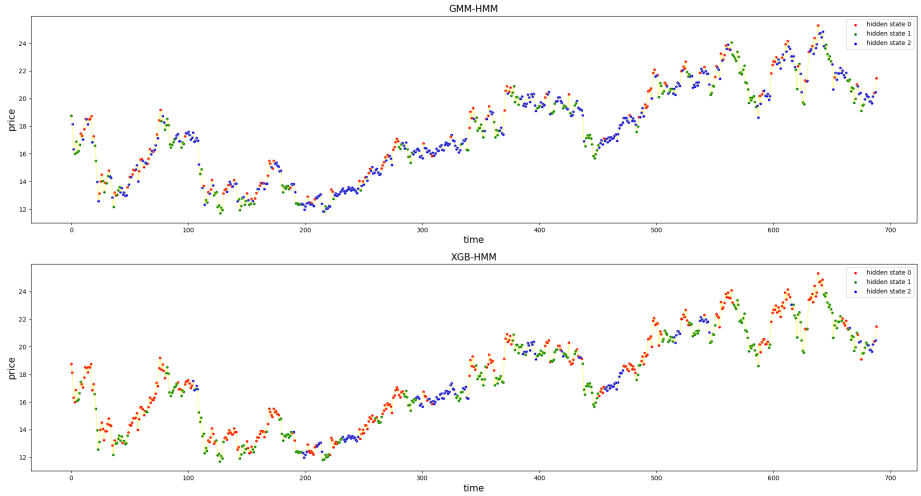


Fig.4.8 GMM-HMM与XGB-HMM测试集对比图

4.4 模型优缺点分析与模型改进

优点；

缺点：在GMM-HMM模型与XGB-HMM模型中，我们使用可视化方法粗略地观察了状态序列S与Y特征之间的关系，认为红色状态代表着上升，绿色状态代表着下降，蓝色状态代表着震荡。

但是根据XGB-HMM的训练集和测试集结果，我们可以看到，三个状态与股票价格上升，震荡，下跌并不十分吻合。例如：当股票价格达到局部极大值，转而开始下降时，此时的状态仍是红色，过了几天才是绿色。

我们小组认为，之前的模型并没有考虑到每一个结点所对应的状态的概率。例如，当结点对应的状态为状态1，我们认为这个结点一定会是状态1。但是根据XGB-HMM模型，我们可以得到每一个结点对应每个状态的概率$P\{S_t = i\}, i = 1, 2, 3$，进而得到矩阵X。

$$X = \begin{pmatrix} P\{S_1 = 1\}.....P\{S_{T-1} = 1\} \\ P\{S_1 = 2\}.....P\{S_{T-1} = 2\} \\ P\{S_1 = 3\}.....P\{S_{T-1} = 3\} \end{pmatrix}$$

下面我们小组使用LSTM模型，进一步X与Y特征之间的关系。

5 GMM-HMM+LSTM模型

5.1 引言

长短时记忆网络（LSTM）算法由Sepp Hochreiter和Jurgen Schmidhuber在Neural Computation上首次公布。

LSTM是一个由X映射到Y的过程，X是一个n × k 维度的矩阵,Y是一个n行的列向量，LSTM可以有多个X对应到同一个Y。

长短时记忆网络(Long Short Term Memory)模型由专门的记忆存储单元组成，通过精心设计的遗忘门、输入门和输出门来控制各个记忆存储单元的状态，通过门的控制保证了随着隐藏层在新的时间状态下不断叠加输入序列，前面的信息能够继续向后传播不消失[6]。

图中使用的符号含义如下：

类别	意义
X	信息
+	增加信息
σ	sigmoid层
tanh	tanh层
h(t-1)	上一个LSTM单元的输出
c(t-1)	上一个LSTM 单元的记忆
X(t)	当前输入
c(t)	新更新的记忆
h(t)	当前输出

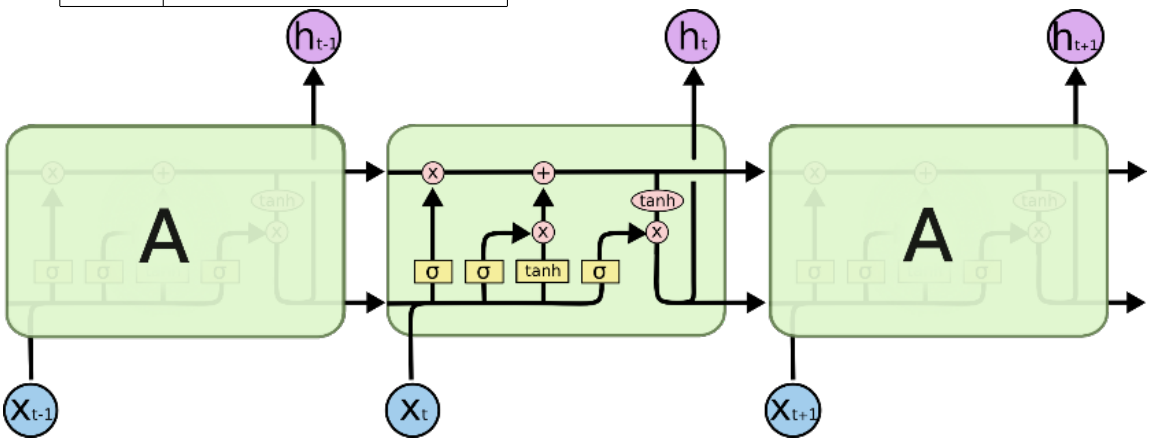
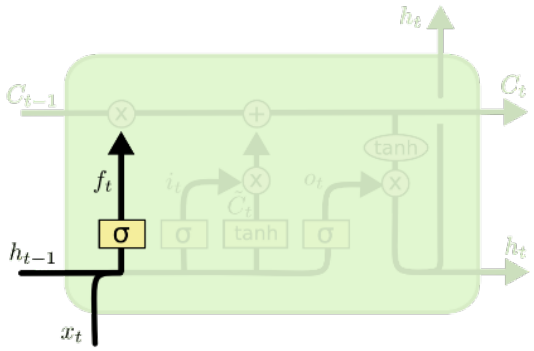


Fig.5.1 LSTM模型结构[4]

长短时记忆网络结构主要由三部分构成：

（1）忘记门（forget gate layer)

忘记门控制着应该忘记哪些信息，通过忘记门的sigmoid 神经层来实现。上一层的输出信息h_{t-1}和当前信息X_t进行线性组合后，利用激活函数，将其函数值进行压缩，得到一个大小在0和1之间的阈值。当函数值越接近1,表示记忆体保留的信息越多。当函数值接近0，表示记忆体丢失的信息越多[3]。忘记门的逻辑设计如下。

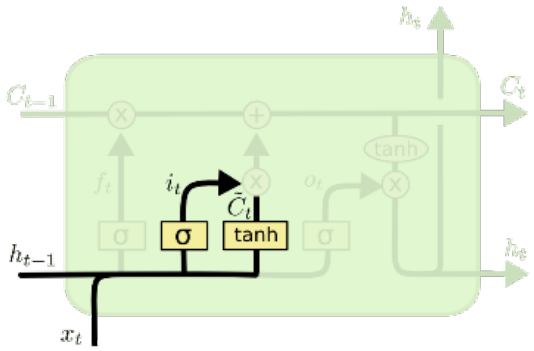


gate.png

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \ + \ b_f\right)$$

Fig.5.1 忘记门的逻辑设计[4]

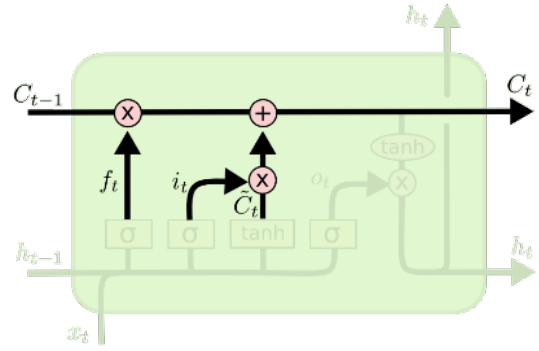
（2）输入门(input gate layer) 输入门决定让多少新的信息加入到单元状态中来。实现这个需要包括两个步骤：首先，输入门的sigmoid 神经层决定哪些信息需要更新；一个tanh 层生成一个向量，作为备选用来更新的内容C_t 。然后，我们把这两部分联合起来，对单元状态进行一个更新[3]。



gate.png

$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \ + \ b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \ + \ b_C)$$

Fig.5.1 输入门与候选门的逻辑设计[4]

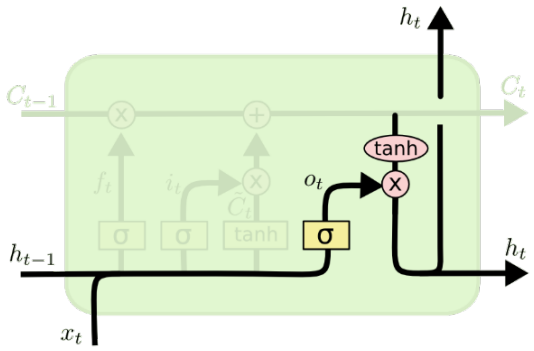


gate2.png

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

Fig.5.1 输入门与候选门的逻辑设计[4]

（3）输出门(output gate layer) 最后，输出门决定输出什么值。这个输出主要是依赖于单元状态C_t，并且还需要经过一个过滤的处理。首先，sigmoid神经层来决定Ct中的哪部分信息会被输出。接着，Ct通过一个tanh 层，把数值都赋值-1 和1 之间，然后把tanh 层的输出和sigmoid 层计算出来的权重相乘，作为最后输出的结果[3]。



$$o_t = \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

gate.png

Fig.5.1 更新后的记忆信息[4]

5.2 训练算法

（1）运行GMM-HMM模型，得到$P\{S_t = i\}, i = \{1, 2, .., N\}, t = \{0, 1, .., T-1\}$

（2）使用概率$P\{S_t = i\}, i = \{1, 2, .., N\}, t = \{0, 1, .., T-1\}$构造出LSTM模型的训练集X。

$$X = \begin{pmatrix} P\{S_1 = 1\}.....P\{S_{T-1} = 1\} \\ P\{S_1 = 2\}.....P\{S_{T-1} = 2\} \\ P\{S_1 = 3\}.....P\{S_{T-1} = 3\} \end{pmatrix}$$

（3）将X和Y特征输入LSTM模型，训练LSTM模型。

待写。。。

5.3 实证结果与分析

我们小组选取的股票为丰原药业，选取2007-01-04至2013-12-17的数据作为训练集，将股票红太阳（000525.XSHE）2014-12-01至2018-05-21的数据作为训练集。

在训练集上训练得到GMM-HMM模型，然后得到数据集的X，然后将这个X作为训练集去训练LSTM模型，记此训练完成的模型为$gmm - hmm + lstm_0$.

然后在测试集上，用之前的$gmm - hmm + lstm_0$模型得到X并输出lstm模型的结果。

5.4 模型优缺点分析与模型改进

6 XGB-HMM+LSTM模型

（1）运行XGB-HMM模型，得到$P\{S_t = i\}, i = \{1, 2, .., N\}, t = \{0, 1, .., T-1\}$

（2）使用概率$P\{S_t = i\}, i = \{1, 2, .., N\}, t = \{0, 1, .., T-1\}$构造出LSTM模型的训练集X。

$$X = \begin{pmatrix} P\{S_1 = 1\}.....P\{S_{T-1} = 1\} \\ P\{S_1 = 2\}.....P\{S_{T-1} = 2\} \\ P\{S_1 = 3\}.....P\{S_{T-1} = 3\} \end{pmatrix}$$

（3）将X和Y特征输入LSTM模型，训练LSTM模型。

6.1 实证结果与分析

我们小组选取的股票为丰原药业，选取2007-01-04至2013-12-17的数据作为训练集，将股票红太阳（000525.XSHE）2014-12-01至2018-05-21的数据作为训练集。

在训练集上训练得到XGB-HMM模型，然后得到数据集的X，然后将这个X作为训练集去训练LSTM模型，记此训练完成的模型为$xgb - hmm + lstm_0$.

然后在测试集上，用之前的$xgb - hmm + lstm_0$模型得到X并输出lstm模型的结果。

6.2 模型优缺点分析与模型改进