AWALEE NOTES

Financial Time Series and Machine Learning: What Are The Advantages of Using Hidden Markov Models?

Study carried out by the Data Science Practice
Special thanks to Pierre-Edouard THIERY

awalee

APRIL 2019

The question of time series modeling is particularly crucial in finance, since the price of an asset is, by definition, a time series. In order to devise promising models, strong mathematical tools are needed to cope with those time series while respecting some of the basic stylized facts of financial time series, such as non-stationarity or non-linearity. Some of those tools are well-known insofar as they have been used for many years; most of them come from the field of econometrics. Nonetheless the rising influence of Machine Learning now entices us to search for new ways of addressing the modeling of time series: that is what we intend to do in this paper, by seeing how Hidden Markov Models (HMM) can be applied when it comes to financial time series modeling.

Financial Time Series: from Econometrics to Machine Learning

Financial times series can easily be considered as the staple of financial analysis: the prices of an asset over a given period of time provide us with a sequence of values which defines such a time series. But, from a mathematical point of view, we would like to go beyond the data and to devise a model which can explain, and in some cases even predict, the evolution of an asset's price.

For many years, this task has been achieved by using econometric methods [1] [2]. We present in this first part the most common methods, before seeing what their limits are.

Definition 1 (Linear Model)
The price of an asset is denoted p_t with $t \in \mathbb{N}$; for instance p_t can refer to a daily observation. A linear model assumes that the price sequence evolves according to the following equation

$$p_t = \alpha + \beta p_{t-1} + \epsilon_t$$

where (ϵ_t) is a noise process, and α and β two parameters we need to estimate.

Under a given hypothesis concerning the noise process, this model consists in a mere linear regression. It is a pretty well-known and easy-to-use model. Nonetheless one of its shortcomings is precisely that it is clearly too simple: it can be useful in some very specific context, but it is unwise to assume that a financial time series can be properly represented via a linear equation. In order to take into account non-linearity, more complex models are possible, for instance:

Definition 2 (Non-Linear Regression)
We keep the same notations as above. We can consider the following non-linear equation:

$$p_t = \alpha p_{t-1} + \beta p_{t-1}^2 + \epsilon_t$$

In order to use those models, be it linear or non-linear, a regression needs to be performed so as to estimate the parameters α and β. The process is rather simple once the form of the equation has been fixed, but this latter question is much more difficult. Indeed financial markets are highly non-linear: catching all the non-linearity within a single equation sounds obviously impossible. But this is only the first of the three main difficulties we can find about econometric models.

The Three Issues
1) It is impossible to sum up all the financial market complexity within one form of equation;
2) A single model cannot work all the time, insofar as the patterns are actually dynamic;
3) A good model should rule out misleading information and be able to make a difference between long-term trends and short-term sideways movements.
To deal with financial time series modeling in a newer and more efficient way than the above-mentioned methods, we present a different framework, based on Hidden Markov Models (HMM).

From Markov Models to Hidden Markov Models

We begin with a few reminders concerning traditional Markov models before delving into the more complex notion of Hidden Markov Models.

Definition 3 (Markov Process)
Let $\mathbb{S}$ be a set of N elements, denoted by their index and called the states:

$$\mathbb{S} = \{1,\ 2, \dots N\}$$

We consider a stochastic process (X_t) with $t \in \mathbb{N}$. For each date t, $X_t \in \mathbb{S}$. X is said to be a Markov chain when the state reached by X at t only depends on the state at the previous step $t-1$. So:

$$\mathbb{P}(X_t = k | X_{t-1},\ \dots,\ X_0)$$
$$= \mathbb{P}(X_t = k | X_{t-1})$$

Moreover the Markov chain is said to be homogenous when the transition probability from one state to another does not depend on t :

$$\mathbb{P}(X_t = k|X_{t-1}) = \mathbb{P}(X_1 = k|X_0)$$

As of now, we denote $A = (a_{ij})_{1 \leq i,j \leq N}$ the matrix of transition probabilities, i.e.

$$a_{ij} = \mathbb{P}(X_1 = j|X_0 = i)$$

For the Markov chain to be fully defined, we need to know the initial probability distribution π across the states. So for $1 \leq i \leq N$:

$$\pi_i = \mathbb{P}(X_0 = i)$$

Therefore a Markov chain X is defined by its initial probability vector π and its transition matrix A.

We provide here a first naive approach so as to model a stock price using a Markov chain. We consider for instance a simple set of states, which aims at describing the main observations in the price movements:
$\mathbb{S} = \{1, 2, 3\}$ where:
- 1 accounts for an upward move of the price;
- 2 accounts for a downward move of the price;
- 3 accounts for a stable price;

Within this framework, if we admit that both π and A are known, it is very easy to compute the probability that the stock price follows a given sequence of states on a given number of days, for example on three consecutive days the sequence $U = \{3, 2, 3\}$:

$$\mathbb{P}(X \text{ follows } U) = \pi_3 \times a_{3,2} \times a_{2,3}$$

Of course, the first difficulty of this model is to obtain the transition matrix A; we do not delve into this problem since such an approach is too naive, and that is why we go a step further and introduce the Hidden Markov Models.

For a Markov model, we directly assimilate the states and the observations, but for a HMM we have to make a difference: we still have a Markov chain, but it is considered hidden, so we do not have access to the hidden states, but for each state we know the probability that a given observation is output. Let us define a HMM formally.

Definition 4 (Hidden Markov Model)
We define $\mathbb{O}$ the set of possible observations, with N_O its cardinal. We have to be very careful with the notations: o_j refers to a generic element within $\mathbb{O}$, and $O(t) \in \mathbb{O}$ is the random variable whose value is the output observed at t.

We also consider X a Markov chain, which plays the role of the hidden Markov chain. We still denote the set of hidden states $\mathbb{S}$, with cardinal N, π and A the initial probability law and the transition matrix. The states $S = \{1, \ldots, N\}$ are not visible, but we know the probability that a given observation o_j is produced when the hidden state is k:

$$b_k(o_j) = \mathbb{P}(O(t) = o_j|X_t = k)$$

So a HMM is fully given by its initial probability law π, its transition matrix A,and its state-output matrix $B \in \mathbb{R}^{N \times N_O}$, which does not depend on time, with:

$$b_{kj} = b_k(o_j) = \mathbb{P}(O(t) = o_j|X_t = k)$$

We denote H a generic HMM:
$H = (\pi, A, B) \in \mathbb{R}^N \times \mathbb{R}^{N \times N} \times \mathbb{R}^{N \times N_O}$

The difference between a Markov model and a HMM is fairly simple: in the first case, we assimilate the states with the possible observations, whereas in the second case we make a difference between the states and the observations and we link the whole with a probability matrix *B*.

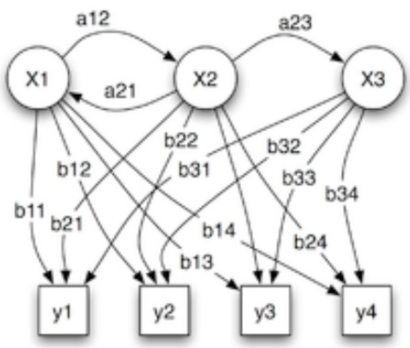

Figure 1: Representation of a simple HMM, with x the hidden states and y the observations

When it comes down to HMM, there are three pivotal questions that need to be addressed:
- 1) Given a HMM *H* and a sequence of *T* consecutive observations denoted $\vec{O} = (O^1, \ldots, O^T)$, how do we compute (efficiently) the probability that such a sequence of observations has been output: $\mathbb{P}\left(\vec{O}|H\right)$?

- 2) Given a HMM *H* and a sequence of *T* observations $\vec{O}$, what is the sequence of hidden states $\vec{S} \in \mathbb{S}$ which is the most likely?

- 3) Given a sequence of observations $\vec{O}$, how do we adjust the HMM *H* in order to maximize $\mathbb{P}\left(\vec{O}|H\right)$?

Hidden Markov Models: how to Solve the Three Questions?

In this part we provide the reader with the three algorithms that enable to solve the three questions [3].

The first question is the simplest one. We consider a given HMM denoted H and a sequence of T observations $\vec{O} = (O^1, \ldots, O^T)$. We would like to compute in an efficient way $\mathbb{P}\left(\vec{O}|H\right)$. Several methods are possible, but as we will see, some are more efficient than others. The intuitive method consists in writing:

$$\mathbb{P}\left(\vec{O}|H\right) = \sum_{\text{all } \vec{S} \in \mathbb{S}^T} \mathbb{P}\left(\vec{O}|\vec{S},\ H\right) \times \mathbb{P}\left(\vec{S}|H\right)$$

and then computing each element of the sum. We do not give all the details of the computation, since the latter has an exponential complexity. We rather explain the forward algorithm, whose complexity is linear.

Definition 5 (Forward Variables)
Considering a vector of T observations denoted $\vec{O} = (O^1, \ldots, O^T)$, the forward variables $\alpha_i(t)$ are defined by:

$$\alpha_i(t) = \mathbb{P}(O(1) = O^1, \ldots,\ O(t) = O^t,\ X_t = i|H)$$

So $\alpha_i(t)$ accounts for the probability of ending with the hidden state i after the first t observations. If we can compute efficiently the forward variables, then the desired probability is:

$$\mathbb{P}\left(\vec{O}|H\right) = \sum_{i=1}^{N} \alpha_i(T)$$

The forward variables can be computed by induction.

Theorem 1 (Computation of the forward variables by induction)
To compute the forward variables we use the following equations:

$$\alpha_i(1) = \pi_i b_i(O^1)$$

$$\alpha_j(t+1) = b_j(O^{t+1}) \sum_{i=1}^{N} \alpha_i(t) a_{ij}$$

The forward procedure is much more efficient than the first naive one: the latter has an exponential complexity with respect to T, i.e. $O(T2^T)$, whereas the forward procedure has a linear complexity with respect to T, $O(TN^2)$.
We now treat the second classic problem concerning Hidden Markov Models: given a sequence of observations $\vec{O}$, what is the most likely sequence of hidden states $\vec{S}$? Again several methods are possible, but we are interested in the most efficient, which is called the "Viterbi Algorithm".

Definition 6 (Viterbi Algorithm)
This algorithm aims at solving the following problem:

$$\arg\max_{\vec{S} \in \mathbb{S}^T} \mathbb{P}\left(\vec{S}|\vec{O},\ H\right)$$

which is equivalent to solving:

$$\arg\max_{\vec{S} \in \mathbb{S}^T} \mathbb{P}\left(\vec{S},\ \vec{O}|\ H\right)$$

To do so in an efficient way, we introduce the δ variables:

$$\delta_j(t) = \max_{(s_1,\ \ldots,\ s_{t-1}) \in \mathbb{S}^{t-1}} \mathbb{P}\left(X_1 = s_1, \ldots,\ X_{t-1} = s_{t-1},\ O^1, \ldots,\ O^t,\ X_t = j|H\right)$$

The computation is done by induction. At step 1, we have:

$$\delta_i(1) = \pi_i b_i(O^1)$$

Then we use the induction equations to compute δ when $t \geq 1$:

$$\delta_j(t+1) = b_j(O^{t+1}) \max_{1 \leq i \leq N} \delta_i(t) a_{ij}$$

When we reach T, we know that the probability and the optimal final hidden states are:

$$P^\star = \max_{1 \leq i \leq N} \delta_i(T)$$

$$s_T^\star = \arg\max_{1 \leq i \leq N} \delta_i(T)$$

We can read the optimal sequence of hidden states from the end to the beginning by using the equation, for $t < T$

$$s_t^\star = \arg\max_{1 \leq i \leq N} \delta_i(t) a_{i s_{t+1}^\star}$$

And so $\vec{S}^\star = (s_1^\star, \ldots,\ s_T^\star)$.

This algorithm, although its formulation seems a bit complicated, is actually rather simple: the δ variables account for the most likely paths of size t that end with a given hidden state. To solve the overall question, we need to compute the maximum of the numbers $\delta_i(T)$ over i. The argmax gives the final hidden state of the optimal path. So the solution resides in the ability of computing the δ variables over t in an efficient way, which is possible by using the induction equations.

The third question is probably the most important one, as it aims at dealing with the learning process. We no longer consider a fixed HMM H, but we would like to modify H, which means π, A and B, in order to maximize the probability of a given sequence of observations $\vec{O}$ under the modified hidden Markov model H:

$$\max_{H\ :\ HMM} \mathbb{P}\left(\vec{O}|H\right)$$

The question is at the center of the HMM theory.

Originally HMM were mainly used for handwriting recognition [4]: each letter, for instance *a*, can be seen as a HMM H_a, and by using many different handwritten *a*, we would like to improve H_a (third question) so that H_a maximizes the probability of a sequence of observations which is similar to the observations contained in the training set (the examples of the handwritten *a*). Then, when reading a text, in order to identify each letter in a word (this letter is seen as a vector of observations $\vec{O}$), we use the first question to compute $\mathbb{P}\left(\vec{O}|H\right)$ for all the HMM *H* we have trained (one for each letter in the alphabet). The HMM which maximizes this probability is considered to be the one associated with the letter, and thus we can recognize the written letter.

To train a HMM, we use the Baum-Welch algorithm. Here we provide the user with the basics of this algorithm. First, we define the backward variables β:

$$\beta_i(t) = \mathbb{P}\left(O(t+1) = O^{t+1}, \dots, \ O(T) = O^T | X_t = i, H\right)$$

Those β variables are very similar to the above-mentioned α variables: we can compute the β very easily by induction, and this computation can be used to solve the first question (backward procedure). The whole computation is based on the following equations:

$$\beta_i(T) = 1$$

$$\beta_i(t) = \sum_{j=1}^{N} a_{ij} b_j(O^{t+1})\beta_j(t+1)$$

Both α and β variables are necessary for the Baum-Welch algorithm. Indeed if we define the following quantities

$$p(t,i,j) = \mathbb{P}\left(X_t = i, \ X_{t+1} = j | \vec{O}, H\right)$$

$$\gamma_i(t) = \mathbb{P}\left(X_t = i | \vec{O}, H\right)$$

they can be rewritten using the forward and backward variables:

$$p(t,i,j) = \frac{\alpha_i(t) a_{ij} b_j(O^{t+1})\beta_j(t+1)}{\mathbb{P}\left(\vec{O}|H\right)}$$

$$\gamma_i(t) = \sum_{j=1}^{N} p(t,i,j)$$

Definition 7 (Baum-Welch Algorithm)
This algorithm aims at solving the following problem, with a given sequence of observations $\vec{O}$:

$$\max_{H \,:\, HMM} \mathbb{P}\left(\vec{O}|H\right)$$

To do so, we repeat the following process in order to reach the optimal values for the new HMM H′

$$\pi_i' = \gamma_i(1)$$

$$a_{ij}' = \frac{\sum_{t=1}^{T-1} p(t,i,j)}{\sum_{t=1}^{T-1} \gamma_i(t)}$$

and for $o \in \mathbb{O}$ a possible observation:

$$b_i(o)' = \frac{\sum_t \mathbf{1}_{O(t)=o}\gamma_i(t)}{\sum_{t=1}^{T} \gamma_i(t)}$$

We do not delve into all the details of those computations, insofar as it would lead us far beyond the scope of this paper. Just two things are worth noticing: Baum has proved that, either H is already the critical point and in this case H′ =H, or $\mathbb{P}\left(\vec{O}|H'\right) > \mathbb{P}\left(\vec{O}|H\right)$, showing *H′* has a greater chance to produce the output $\vec{O}$. To learn more about this algorithm, the reader can have a look at *Markov Models For Pattern Recognition*, by Gernot A. Fink [3].

HMM and Financial Time Series

In this last part, we suggest a simple way to use HMM to deal with financial time series by presenting a pedagogical model. We remind the reader the staple of HMM is that we make a difference between a hidden state, which is not visible, and an observation, which is visible but which does not give direct access to the hidden state.

We would like to identify three different regimes for a given asset: when there is a positive trend, a negative trend, or a flat period. To do so we considerer three HMM, one for each regime: H^+ for the positive trend, H^- for the negative trend, and $H^=$ for the flat period. For our three HMM and to keep our analysis as simple as possible, we consider a set $\mathbb{O}$ made of three different observations based on the price $(p_t)_{t\in\mathbb{N}}$

- $O(t) = o_1$ accounts for a price going up, for instance when $\frac{p_t - p_{t-1}}{p_{t-1}} > \mu$ with µ a fixed threshold;

- $O(t) = o_2$ accounts for a price going down, for instance when $\frac{p_t - p_{t-1}}{p_{t-1}} < \mu$;

- $O(t) = o_3$ accounts for a stable price, when none of the two above-mentioned conditions is fulfilled.

Let us denote *N* the number of hidden states. We choose an horizon *T* for a sequence of observations. At each date *t* (each day), we consider a sequence of *T* observations $\vec{O} = (O(t-T+1), \dots, O(t))$.

Then, by computing the probability of observing such a sequence for the three HMM H^+, H^- and $H^=$, we find the HMM which is the most likely to characterize the current period. This way we would like to know how to describe the current period (positive trend, negative trend, or flat period), and then adjust, for instance, our investments.

Nonetheless, to do that, it is necessary to train our HMM: we would like indeed that H^+ for instance is good at identifying periods with a positive trend. To perform the training step of our HMM, we consider a training period $[0, T_{tr}]$, and for each day within this period, we know the price of the asset: $(p_t)_{t \in [0, T_{tr}]}$. Then, if we focus only on the training of H^+, we have to identify across the training period the sequences of 10 consecutive days we consider to be a period with a positive trend. Let us say we have identified such a period: $[t^+_{initial}, \ldots, t^+_{initial} + 9]$, with $0 \leq t^+_{initial} \leq t^+_{initial} + 9 \leq T_{tr}$.

The next step consists in defining the vector

$$\vec{O}^+ = (O(t^+_{initial}), \ldots, O(t^+_{initial} + 9))$$

We use it as the sequence of the observations of the Baum-Welch algorithm: we define H^+ as the Hidden Markov Model which maximizes the probability of having such a sequence of observations. Since this algorithm needs a "starting" HMM which will be adapted, we can choose to start with a conventional HMM, for instance with uniform laws for π, A and B.

We have implemented this model on a simple numerical example. We chose to set the meta-parameters of our model to the following values: $N = 5$, $T = 10$ days. We work with a set of 200 observations of a simulated price; we generated those data using a classic log-normal distribution of the price, using various trends. As explained above, we divide our data into two sets: a training set, and then a test set.

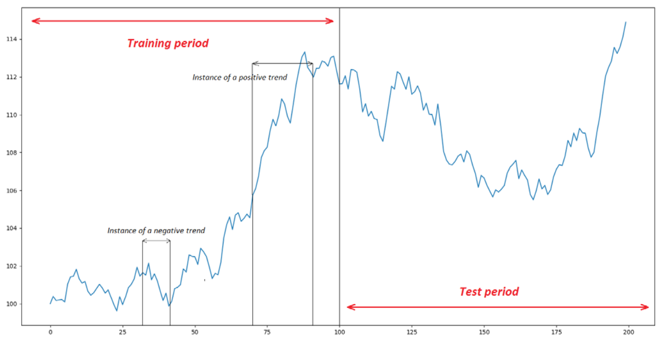

Figure 2: The data for our numerical example

For each t within the test period, we have considered the vector of T observations for the past T days; then computing the probabilities to observe this sequence for our three HMM, we detect the most likely trend at t. The results are shown in the following graph:

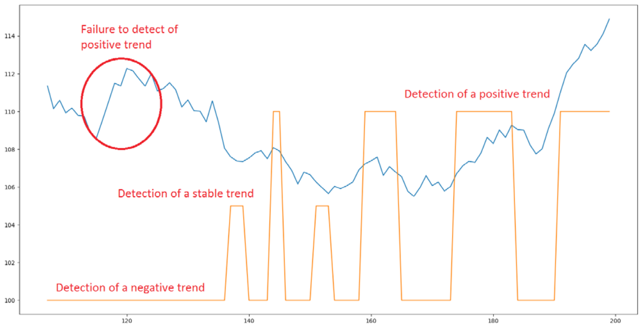

Figure 3: The detection of trend for the test period

The bottom line, with value 100, accounts for a negative trend; the intermediary line, with value 105, accounts for a stable trend; the top line, with value 110, accounts for a positive trend. As we can see, the negative trends, especially the one from date $t = 100$ to $t = 140$, are rather well detected. The stable periods are the most difficult to detect, since they often lie between two trends; it would have been easier if we had had longer stable periods. The positive trends also appear on our graph, especially around the date $t = 180$ and after $t = 190$.
If those results seem encouraging, we also see that the model can fail to detect a specific trend, for instance the positive trend around $t = 120$. Such a failure is easily understandable: as we have mentioned previously, this model is a pedagogical one. Of course when dealing with true financial data, it should be strengthened to make more efficient estimations. We mention here two trails among many: how to optimally choose the meta-parameters? How to replace the finite set of observations by a continuous one, so as to have a less naive view of the market?

The study of financial time series is of the greatest importance in finance. The traditional methods have been well-known for many years and come mainly from the field of econometrics. Nonetheless, those econometric approaches face many difficulties, and that is why we are interested in proposing a new framework so as to study financial time series with Machine Learning techniques, especially Hidden Markov Models.
Those models are derived from classic Markov chains, but aim at differentiating the states, which are no longer visible, and the observations, which are visible but also an imperfect representation of the hidden state. HMM form a theory that has been known for many years, for instance due to its application for handwriting recognition, and thus some efficient algorithms exist to solve the three problems which are ubiquitous within this theory.

The main result is the Baum-Welch algorithm, which is used to optimize the components of a HMM so as to make the latter more representative of a series of observations.

We have presented a first approach of how to use HMM when it comes to financial time series. We have tried to keep it as simple as possible, especially for pedagogical purposes, but it is possible to go further. Some improvements are possible, for instance to replace the finite set of observations by a continuous one.
Nonetheless, if we would like to go beyond financial time series, it is worth noticing that we can find other applications for HMM in the field of finance. For instance they can be used to tackle volatility and regime issues, and then be applied within a systematic strategy framework. HMM, like many other Machine Learning techniques, are now forming an expanding area in finance.

REFERENCES

[1] Stephen Taylor. Modelling Financial Times Series. World Scientific Pub Co Inc, 1986.

[2] Terence C. Mills and Raphael N. Markellos. The Econometric Modelling of Financial TImes Series. Cambridge University Press, 2008.

[3] Gernot A. Fink. Markov Models for Pattern Recognition. Springer, 2007.

[4] B. S. Saritha and S. Hemanth. An efficient hidden markov model for offline handwritten numeral recognition. InterJRI Computer Science and Network- ing, page 9, jan 2010.

ABOUT US

Awalee est un cabinet de conseil indépendant spécialiste du secteur de la Finance, créé en 2009 et qui compte plus de 80 collaborateurs.

Nous sommes en mesure à la fois d'adresser des sujets relatifs à l'expertise des métiers de la Finance (Consulting) et de conduire des projets d'organisation et de transformation (Advisory). Et nous le faisons grâce à la synergie agile de ces deux savoir-faire.

Nos expertises s'exercent dans la conformité réglementaire, la finance quantitative, la fonction finance, la gouvernance des outils & systèmes, le management des risques et les marchés financiers. Au-delà de ce que nous faisons, il y a comment nous le faisons : viser l'excellence et repousser nos limites tout en cultivant la convivialité et en favorisant l'esprit d'équipe.

Nous sommes Awalee : nous sommes AWARE & AWESOME.

Awalee consulting
77 Boulevard Berthier
75017 Paris

www.awaleeconsulting.com

linkedin.com/awaleeconsulting