Expectation Maximization Algorithm for Weighted k-components Gaussian Mixture: An Application to High Frequency Financial Data.

Abstract

Risk management is one of the key factors in the financial markets in general and derivative pricing industry in specific. Estimation of risk in finance requires flexible models capable of giving the best fit to financial data, especially high-frequency data, to prevent under-estimation or over-estimation of financial risk. The aim of this research is to use a flexible Gaussian mixture in modeling high-frequency financial data. The Expectation Maximization algorithm is used in estimating the parameters of the Gaussian mixture model under the assumption of missing information. The research uses high-frequency financial data from Chinese Soy Beans Futures which were captured at a minute interval from 9/17/2013 8:59:00 AM to 2/1/2021 10:59:01 PM. The iterative process of the Expectation Maximization algorithm for a continuous probability density is shown and hence, for a two-component Gaussian mixture. The generalization has been given for a k-component mixture with probability w_i where i is the i-th component. The flexible distributions have been applied to the data each time with increasing component starting with two components. Selection criterion is based on the BIC values which were found to ignore over-fitting. The BIC gradient were computed and used as selection criterion instead. We found that beyond five components, if we increase the number of components, the BIC gradient scores remains almost constant. Therefore, there is no gain in the mixture model by increasing the number of components. As a result, we conclude that the best model is the one with four components. Future research is however required to come up with the best model under BIC criterion with penalty for over-fitting as the model complexity increases.

1 Introduction

The Expectation Maximization algorithm was first developed by,Dempster et al. (1977) to estimate the maximum likelihood estimates where we have missing data or latent variables. It is an iterative approach applicable in most scenarios where weighted statistical models deal with incomplete data. In their papers, Dempster and Laird et al. (1987) defines the term "incomplete data" as the existence of two sample spaces X and Y and a many-one mapping from X to Y .

In Sammaknejad et al. (2019), the Expectation Maximization (henceforth, EM Algorithm) has been applied before in process identification using data-driven approach since the algorithm ensures convergence of the likelihood function as a result of its monotonic property. Unknown time delay, sensor mulfunction and multiple process operation conditions are some examples of missing observation sources in this paper. The EM Algorithm helps in designing identification algorithms used in such identification processes.

The Gaussian model is one of the most used modeling distribution in statistical analysis. This model has been used for a very long time by many authors because of its simple parameters used in population description and other areas such as classroom performance, weights, and heights. The finite mixtures of Gaussian have also gained popularity in the field of machine learning and artificial intelligence for clustering, data mining, and image recognition purposes, Viroli and McLachlan (2019). The Gaussian mixture is well known limiting case for any continuous distribution using its normal densities. We notice that Chen (1995) demonstrates that when the number of components in a finite mixtures of Gaussian is known, the optimal rate of convergence is faster, otherwise the rate is slower. When deciding the number of such components, we need to take a lot of care since too many components tend to overfit the data while few tend to underfit the data, Huang et al. (2017). What is therefore more important is the selection of an optimal number of components of normal densities in a Gaussian mixture in practical applications.

2 Literature Review

Financial modeling has been an area of interest to researchers in the financial field for a long time, with the use of hyperbolic models gaining momentum in the field, Oduor and Anyango (2024). In Huang et al. (2017), a proposed modified EM algorithm was used to discard insignificant components, split any large components, and prevent the generation of any large components in the Gaussian mixture model. This is to obtain an optimal number of components for a finite Gaussian mixture.

With the new technological advances, the introduction of complex systems has enabled easy and seamless collection and storage of financial market data at very short time intervals. This high frequency basis has enabled the develop-

ment of powerful tools used in analysis of such big data. These data include stock price, futures, exchange rates, interest rates, which can be collected at a frequency as high as every second, minute, hour, or day, Goodhart and O'Hara (1997). Modeling this kind of data has been the interest of researchers within the financial scope with aims of risk management and portfolio allocation. Cao et al. (2019), proposed the use of hybrid model that combined Long-Short Term Memory (LSTM) and Complete Ensemble Empirical Mode with Adaptive Noise to improve accuracy for stock market price forecasting. The model was later tested on global stock market indices and was reported to show good performances. The data used in the paper was however daily stock price indices whose frequency was a little bit lower.

The use of machine learning tehniques such as the Artificial Neural Networks and Random Forest have been proposed by Vijh et al. (2020) and Convolusional Neural Networks, Deep Beliefs Networks by Sezer et al. (2020) in modeling and forecasting of daily financial time series data have been evident in the literature. The LSTM and ARIMA models have also been used in modeling and forecasting financial data recorded at daily intervals, Moghar and Hamiche (2020), Siami-Namini et al. (2018), and Siami-Namini et al. (2019).

Even though the Gaussian Mixture models have been rampant in the field of machine learning, it has solely been applied to fields such as clustering analysis, unsupervised anomaly detection as deep auto-encoding finite mixture model Zong et al. (2018), in topology as a sub-manifold of probability densities in optimal mass transport network, Chen et al. (2018). Other authors have used the GM in foreground video surveilence through background subtraction, Goyal and Singhai (2018), emotion recognition using hybrid GM and deep neural network, Shahin et al. (2019) as well as outliers detection in seasonal univariate network traffic, Reddy et al. (2017).

The use of finite GM has therefore not been extensively applied to the fields of financial time series modeling especially where data is in high frequency, say "minutely" with missing information, incomplete information. It is in the interest of this paper that a mixture of Gaussian components can very well capture the peaks of high-frequency data in the financial stock market. This study intends to bridge the gap in the scientific literature by extensively exploring the capabilities of the finite Gaussian mixtures in high-frequency financial data using the EM Algorithm approach. This concept will therefore be used in risk management as it explores the behavior of the derivative markets.

3 Research Methodology

3.1 The E-M Algorithm Framework

This is a two-step iterative proccess that aims at maximizing the log-likelihood function during parameter estimation. The steps are; Expectation (E-step) and the Maximization (M-step) (as in Dempster et al. (1977)).

In the E-step we choose a parameter ϕ that maximizes the likelihood but this

is hard because we have missing data. To find ϕ that maximizes the likelihood:

- We will try to fill in the missing data.
- Then we find ϕ that maximizes the likelihood.
- Repeat this process until we are happy with our answer.

Let X be observed data and ϕ the parameter we wish to estimate. The likelihood is obtained as follows:

$$L(\phi) = \log \mathbf{f}(x|\phi) \tag{1}$$

where f(x—ϕ) is the probability density function of X given ϕ.
The problem is to maximize the log-likelihood log f(x—ϕ). But as we earlier noted, this is difficult but we can consider the completed data and the current estimates of the parameters. Let us consider this as follows:

$$L(\phi) = \log \mathbf{f}(x|\phi) = \int_c \mathbf{f}(x, c|\phi)dc \tag{2}$$

where $\mathbf{f}(x, c|\phi)$ is a joint distribution, C is complete data which contains both the original data X and the missing data. Integrating out c we have:

$$L(\phi) = \log \int_c \mathbf{f}(c|\phi)dc \tag{3}$$

this is because C contains X, we therefore ignore X.
Now let $\hat{\phi}^{(t)}$ be the estimate of the parameter ϕ at time t. We can rewrite equation 3 as follows:

$$L(\phi) = \log \int_c \frac{\mathbf{f}(c|\phi)}{\mathbf{f}(c|x, \hat{\phi}^{(t)})} \mathbf{f}(c|x, \hat{\phi}^{(t)})dc \tag{4}$$

We note that the two equations are the same, except for a modification of equation 3 by introducing a constant,

$$\frac{\mathbf{f}(c|x, \hat{\phi}^{(t)})}{\mathbf{f}(c|x, \hat{\phi}^{(t)})} = 1$$

which does not change it's meaning. The equation 4 is a log of expectation because we have a quantity,

$$\frac{\mathbf{f}(c|\phi)}{\mathbf{f}(c|x, \hat{\phi}^{(t)})}$$

and a probability,

$$\mathbf{f}(c|x, \hat{\phi}^{(t)})dc$$

Integrating this over all c gives an expectation as follows,

$$L(\phi) = \log E_{c|x,\hat{\phi}^{(t)}} \left[\frac{\mathbf{f}(c|\phi)}{\mathbf{f}(c|x, \hat{\phi}^{(t)})} \right]$$

At this point, we have the log of expectation of some function. We therefore apply the Jensen's inequality to obtain the lower bound of the likelihood function.

3.2 Jensen's Inequality

For a concave probability function f,

$$f(E(x)) \geq E(f(x))$$

see Liao and Berg (2019) for details.
We can therefore apply this approach to our problem above. Since our log is concave;

$$L(\phi) = \log E_{c|x,\hat{\phi}^{(t)}} \left[\frac{f(c|\phi)}{f(c\ x,\ \hat{}^{(t)} \mid \phi\ )} \right]$$

$$\geq E_{c|x,\hat{\phi}^{(t)}} \left[\log \frac{f(c\ \phi)}{f(c\ x,\ \hat{}^{(t)} \mid \phi\ )} \right] \quad (5)$$

Equation 5, which is an expectation of log, is our lower bound for $L(\phi)$
By properties of logarithm, we can rewrite,

$$\log \frac{f(c|\phi)}{f(c|x,\ \hat{\phi}^{(t)})}$$

as,

$$\log f(c|\phi) - \log f(c|x,\ \hat{\phi}^{(t)})$$

We can then rearrange 5 as follows;

$$E_{c|x,\hat{\phi}^{(t)}}[\log f(c|\phi)] + E_{c|x,\hat{\phi}^{(t)}}[-\log f(c|x,\ \hat{\phi}^{(t)})] \quad \forall\ \phi \quad (6)$$

This equation can be simplified in two parts as follows;

$$G(\phi|\hat{\phi}^{(t)}) + I(c,\ x,\ \hat{\phi}^{(t)}) \quad (7)$$

where the G function is called the objective function while the I function is the differential entropy. Since the I function is not a function of ϕ, it will not be used as part of the log-likelihood equation. The lower bound of the log-likelihood can therefore be written as;

$$L(\phi) \geq G(\phi|\hat{\phi}^{(t)}) + I(c,\ x,\ \hat{\phi}^{(t)}) \quad (8)$$

In the M-step, we work to improve the G function which will improve the $L(\phi)$. The objective of E-M algorithm is to improve the $L(\phi)$, but this cannot be done directly since we have missing data. If we consider a point on the curve where $\phi = \hat{\phi}^{(t)}$, this expectation becomes a constant and the inequality becomes an equality. Therefore, at $\phi = \hat{\phi}^{(t)}$,

$$L(\phi) = G(\hat{\phi}^{(t)}|\hat{\phi}^{(t)}) + I(c,\ x,\ \hat{\phi}^{(t)}) \quad (9)$$

By subtracting equation 9 from 8, we can sufficiently say that,

$$L(\phi) - L(\hat{\phi}^{(t)}) \geq G(\phi|\hat{\phi}^{(t)}) - G(\hat{\phi}^{(t)}|\hat{\phi}^{(t)}) \quad (10)$$

We choose ϕ that maximizes the objective function $G(\hat{\phi}^{(t)}|\hat{\phi}^{(t)})$ and therefore equation 10 only holds because $G(\phi|\hat{\phi}^{(t)}) - G(\hat{\phi}^{(t)}|\hat{\phi}^{(t)}) \geq 0$ and therefore $L(\phi) - L(\hat{\phi}^{(t)}) \geq 0$ We maximize the G function as follows;
Let $\hat{\phi}^{(t+1)}$ be the estimate of ϕ that maximizes the G function.

$$\hat{\phi}^{(t+1)} = \text{argmax}_{\phi} G(\phi|\hat{\phi}^{(t)}) \tag{11}$$

By this definition,
$G(\hat{\phi}^{(t+1)}|\hat{\phi}^{(t)}) \geq G(\hat{\phi}^{(t)}|\hat{\phi}^{(t)})$
and therefore,
$L(\hat{\phi}^{(t+1)}) \geq L(\hat{\phi}^{(t)})$. This implies that the likelihood function will keep increasing as the iteration process continues.
In summary, the general procedure of the E-M algorithm is as follows:

- We pick an initial parameter value ϕ^0 based on prior knowledge of what it might be.
- Through iteration, we improve the estimate of ϕ through two steps. The Expectation step computes the objective function G and the maximization step re-estimates ϕ by maximizing the objective function G.
- The two steps are repeated over and over again until the likelihood $L(\phi)$ converges(until there is no more change in the value of the parameter estimate).
- Stopping criterion. If we arrive at a new parameter whose value is smaller than a threshold specified in advance, ϵ,
  $|\hat{\phi}^{(t+1)} - \hat{\phi}^{(t)}| < \epsilon$,
  stop the algorithm, else, return to step 1.

3.3 The E-M Algorithm for k-component Gaussian Mixtures

Let $X = x_1, ..., x_n$ be the observed iid normally distributed random variables. The k-weighted normal mixtures is of the form;

$$f(x_i;\phi) = w_1\Phi(x_i;\mu_1,\sigma_1^2) + w_2\Phi(x_i;\mu_2,\sigma_2^2) + ... + w_k\Phi(x_i;\mu_k,\sigma_k^2) \tag{12}$$

where w_j are the weights of the components in the mixture and $\phi = (w_j, \mu_1, \sigma_1^2\mu_2, \sigma_2^2, ..., \mu_k, \sigma_k^2)$ are the parameters of the k components Gaussian mixture. For simplicity, let us observe $k = 2$ component Gaussian mixture. We therefore have the following likelihood;

$$L(\phi; x) = \prod_{i=1} w\Phi(x_i;\mu_1,\sigma_1^2) + 1 - w\Phi(x_i;\mu_2,\sigma_2^2)$$

We can formulate the log-likelihood as follows;

$$\log L(\phi;\, x) = \sum_{i=1} \log[w\Phi(x_i; \mu_1, \sigma_1^2) + 1 - w\Phi(x_i; \mu_2, \sigma_2^2)]$$

Since we have an addition sign inside the logarithm bracket, computing this log-likelihood directly from here might be difficult. We also note that we have no information from which component did each observed data come from. If we did, we would simply fit normal distributions to each group.
Let us define the complete dataset where $X = x_1, ..., x_n$ and $Z = z_1, ..., z_n$ are observed and latent variables respectively such that;
$Z_i = 1$ if x_i belongs to the first component and $Z_i = 0$ if $x_i = 1$ belongs to the second component. The likelihood of the complete data is;

$$L(\phi|x,\, z) = \prod_{i=1} [w\Phi(x_i; \mu_1, \sigma_1^2)]^{Z_i} [1 - w\Phi(x_i; \mu_2, \sigma_2^2)]^{1-Z_i}$$

The log-likelihood therefore is;

$$\log L(\phi|x, z) = \sum_{i=1} Z_i \ \log w + \log\Phi(x_i; \mu_1, \sigma_1^2) \ + \sum_{i=1} 1-Z_i \ \log(1-w) + \log\Phi(x_i; \mu_2, \sigma_2^2) \tag{13}$$

This equation (13) is the G function, $G(\phi|\hat{\phi}^t)$.

3.3.1 The Expectation Step

In this step, we work to find the expected value with respect to the latent variables. This is done as follows;

$$G(\phi|\phi^t) = E_z \ \log L(\phi|x, z)|x_i, \phi^t$$

substituting with the log-likelihood equation we have;

$$G(\phi|\phi^t) = E_z \left[\sum_{i=1} Z_i \ \log w + \log\Phi(x_i; \mu_1, \sigma_1^2) \ + \sum_{i=1} 1-Z_i \ \log(1-w) + \log\Phi(x_i; \mu_2, \sigma_2^2) \ |x, \phi^t \right]$$

We are taking the expected values of z and therefore, the following are constants;

$\log w + \log\Phi(x_i; \mu_1, \sigma_1^2)$ and $\log(1-w) + \log\Phi(x_i; \mu_2, \sigma_2^2) \ |x, \phi^t$, therefore their expected values remain constant.

$$G(\phi|\phi^t) = \sum_{i=1} E[Z_i|x, \phi^t] \ \log w + \log\Phi(x_i; \mu_1, \sigma_1^2) \ + \sum_{i=1} 1-E[Z_i|x, \phi^t] \ \log(1 - w) + \log\Phi(x_i; \mu_2, \sigma_2^2) \ |x, \phi^t$$

To complete this step, we have to compute the expectation for $i = 1, ..., n$,

$$E[Z_i|x, \phi^t] = 1 * P\,r[Z_i = 1|x, \phi^t] + 0 * P\,r[Z_i = o|x, \phi^t]$$

This gives,

$$Pr[Z_i = 1|x, \phi^t]$$

which can also be written as,

$$E[Z_i|x, \phi^t] = \frac{f(x_i|Z_i = 1,\phi^t)Pr(Z_i = 1|\phi^t)}{f(x_i|\phi^t)}$$

Therefore we have,

$$E[Z_i|x, \phi^t] = \frac{\Phi(x_i|\mu_1^t,\sigma_1^{t2}) * w_i^{(t)}}{w_i^{(t)}\Phi(x_i|\mu_1^t, \sigma_1^{t2}) + (1 - w_i^{(t)})\Phi(x_i|\mu_2^t, \sigma_2^{t2})}$$

which can simply be put as;

$$E[Z_i|x, \phi^t] = \hat{w_i}^{(t)}$$

for $i = 1, ..., n$ Therefore, by substitution, the objective function becomes,

$$G(\phi|\phi^t) = \sum_{i=1}^{n} \left\{ \hat{w}_i^{(t)} \left(\log w + \log \frac{1}{\sqrt{2\pi\sigma_1^2}} \exp\left(-\frac{1}{2\sigma_1^2}(x_i - \mu_1)^2\right) \right) \right\}$$
$$+ \sum_{i=1}^{n} \left\{ (1 - \hat{w}_i^{(t)}) \left(\log(1 - w) + \log \frac{1}{\sqrt{2\pi\sigma_2^2}} \exp\left(-\frac{1}{2\sigma_2^2}(x_i - \mu_2)^2\right) \right) \right\}$$

By opening brackets and expanding, we can simplify this to;

$$G(\phi|\phi^t) = \sum_{i=1}^{n} \left\{ \hat{w}_i^{(t)} \left(\log w - \frac{1}{2}\log 2\pi - \frac{1}{2}\log\sigma_1^2 - \frac{1}{2\sigma_1^2}(x_i - \mu_1)^2 \right) \right\}$$
$$+ \sum_{i=1}^{n} \left\{ (1 - \hat{w}_i^{(t)}) \left(\log(1 - w) - \frac{1}{2}\log 2\pi - \frac{1}{2}\log\sigma_2^2 - \frac{1}{2\sigma_2^2}(x_i - \mu_2)^2 \right) \right\} \quad (14)$$

This equation (14) is the expectation of the objective function G that we shall use to update the parameters in the next step.

3.3.2 The Maximization Step

In this stage, we take the partial derivative of the objective function with respect to each of the four parameters. We estimate the parameters by equating the partial derivatives to zero.

$$\frac{\partial G}{9} = 0$$

$$\partial(w, \mu_1, \sigma_1^2 \mu_2, \sigma_2^2)$$

with respect to each parameters, we have,

$$\frac{\partial G}{\partial w} = \frac{1}{w} \sum_{i=1} \hat{w_i^{(t)}} - \frac{1}{1-w} \sum_{1=1} (1 - \hat{w_i^{(t)}}) = 0$$

we can rewrite this as,

$$(1-w)\sum_{i=1}^{n} w_i^{(t)} - w\sum_{i=1}^{n}(1-w_i^{(t)}) = 0$$

solving for w we have the following updating equation for the weight component.

$$w^{(t+1)} = \frac{1}{n}\sum_{i=1}^{n} w_i^{(t)}$$

The same procedure is carried out for the remaining parameters and their updating equations are as follows,

$$\mu_1^{(t+1)} = \frac{\sum_{i=1}^{n} w_i^{(t)} x_i}{\sum_{i=1}^{n} w_i^{(t)}}$$

$$\mu_2^{(t+1)} = \frac{\sum_{i=1}^{n}(1-w_i^{(t)})x_i}{\sum_{i=1}^{n}(1-w_i^{(t)})}$$

$$\sigma_1^{(t+1)\ 2} = \frac{\sum_{i=1}^{n} w_i^{(t)}(x_i-\mu_1^{(t+1)})}{\sum_{i=1}^{n} w_i^{(t)}}$$

$$\sigma_2^{(t+1)\ 2} = \frac{\sum_{i=1}^{n}(1-w_i^{(t)})(x_i-\mu_1^{(t+1)})}{\sum_{i=1}^{n}(1-w_i^{(t)})}$$

where $\mu_1^{(t+1)}$ and $\sigma_1^{(t+1)\ 2}$ are the updated equations for the mean and the variance of the first component, while $\mu_2^{(t+1)}$ and $\sigma_2^{(t+1)\ 2}$ are the updating equations for the mean and variance of the second component. Therefore, the M-step is complete.

Using these updating equations, we go back to the E-step update the parameters, and proceed again to the M-step. this process is called iteration and it continues until convergence. Since the log-likelihood increases on each iteration, we keep rotating between the two steps until there is no more change in the likelihood. At this point, we will have achieved the best parameter estimators.

For a four-component Gaussian mixture model, i.e, for k = 4, we have;

$$f(x) = \sum_{k=1}^{4} w_k f(x;\phi_k)$$

as long as $\sum_{k=1}^{4} w_k = 1$.

Therefore;

$$\mathbf{f}(x) = \sum_{k=1}^{\infty} w_k \Phi(x; \mu_1, \sigma_1^2, \mu_2, \sigma_2^2, \mu_3, \sigma_3^2, \mu_4, \sigma_4^2)$$

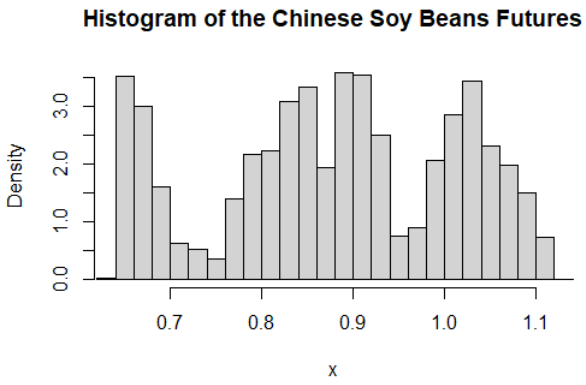
From here, the E-M steps remain as above.

4 Data Analysis and Results

In this section, we use the weighted k component Gaussian Mixtures in modeling financial data occurring every minute. The data used is obtained from the Soy Beans Futures from the Chinese financial market. Soy Beans futures are financial contracts that grant a buyer the obligation to purchase Soy beans or a seller to sell Soy Beans at a predetermined price and date in the future. The data is captured from 9/17/2013 8:59:00 AM to 2/1/2021 10:59:01 PM at a minute interval.

4.1 Histogram of the Soy Beans Futures

The following graph represents a plot of the histogram of the Chinese Soy Beans Futures.



4.2 E-M Algorithm Parameter Estimates

4.2.1 k = 2-Components Gaussian Mixture

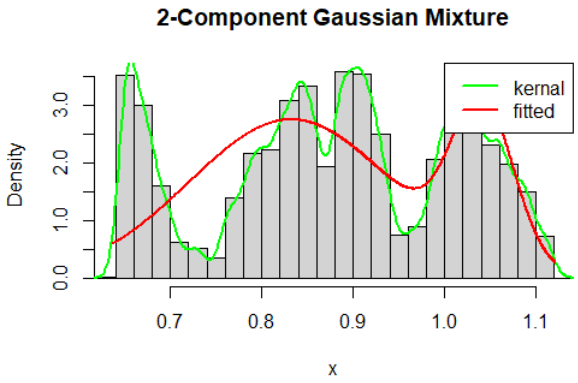
The data is presumed to be captured by two-component Gaussian mixtures with the following properties;

Table 1. k = 2 Component Parameter Estimates

Components	$\hat{\mu}$	$\hat{\sigma^2}$	w_i
Component 1	0.8319752	0.1114532	0.770054
Component 2	1.042019	0.03374	0.229946

From the above table, we notice that approximately 77% of the Soy Beans Futures data comes from the first component of the Gaussian mixture with

mean of 0.8319752 and variance of 0.1114532. The remaining 23% comes from the second component with mean of 1.042019 and variance of 0.03374.
The following plot provides the fitted density and the kernel density of the mixture.



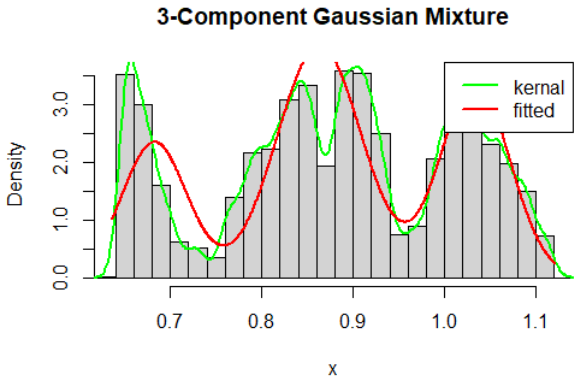
4.2.2 k = 3-Components Gaussian Mixture

Here, the assumption is that the data is captured by three components Gaussian Mixture distribution with the following properties;

Table 2. k = 3 Components Parameter Estimates

Components	$\hat{\mu}$	$\hat{\sigma^2}$	w_i
Component 1	0.6828147	0.03563177	0.2101411
Component 2	0.8633565	0.04628462	0.46767
Component 3	1.033619	0.0380815	0.32219

From the table 2 above, we find that approximately 21% of the data comes from the first component, 47% from the second component, and 32% from the third component of the Gaussian Mixture.
The graph below provides the fitted and kernel of the mixture.



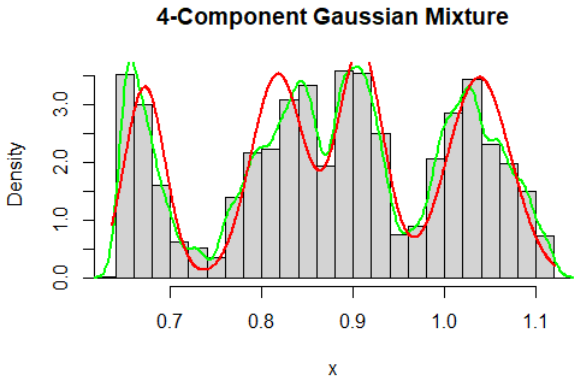
4.2.3 k = 4-Components Gaussian Mixture

The assumption here is that the high-frequency financial data is captured by four-component Gaussian Mixture with the following properties;

Table 3. k = 4 Components Parameter Estimates

Components	$\hat{\mu}$	$\hat{\sigma^2}$	w_i
Component 1	0.6725753	0.0224159	0.186337
Component 2	0.8180116	0.029404	0.26056
Component 3	0.9085797	0.025099	0.249615
Component 4	1.037975	0.034785	0.303484

From the table 3, we notice that approximately 18.6% of the data came from the first component, 26.1% from the second component, 25% from the third component, and 30.3% from the fourth component.
The graph below shows the kernel and the fitted densities of the mixture.



4.3 Goodness of Fit Test

We test how well the mixture model captures the high-frequency financial data using the Bayesian Information Criterion. This is the most preferred method because it penalizes models with a high number of clusters therefore preventing over-fitting. We fit 13 models each time with an additional component starting with one component and BIC values recorded. The model with the least value of BIC will be the one with an optimal number of components. The results are shown below;

Table 4. BIC Values

Number of Components	BIC
1	1089448.0
2	1082600.8
3	1082717.7
4	1077934.7
5	1080019.7
6	1076824.3
7	1054171.3
8	1049752.4
9	1047100.4
10	1040512.3
11	1020806.9
12	821338.7
13	730082.0

From table 4.above, we find that as the number of components increases, the BIC values reduce. This means that the more the number of components the better the model. Even though we have to be very cautious not to have a model that over-fits the data, the BIC in this case seems to have ignored the possibility of over-fitting the data as the complexity of the mixture models increases. We, therefore, cannot decide certainly that the best model is the one with the highest number of components, therefore we ignore this approach and look at its extension.

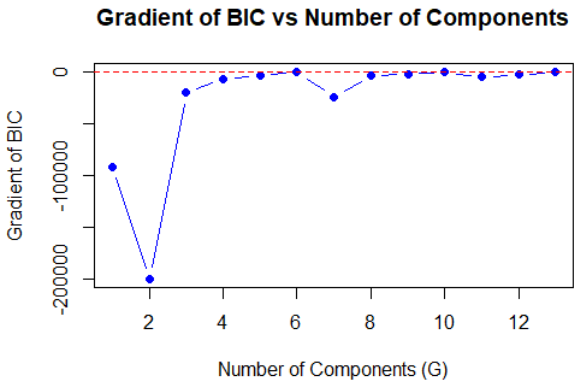
4.3.1 The BIC Gradient

We calculate the gradient of the BIC scores. The simple rule of calculating the gradient of a curve applies here. In addition, if two points on the BIC curve have the same values, then their gradient is zero. Otherwise, if the two points have different values, their gradient can either be positive or negative depending on which value is bigger, the first or second. The calculated values are as follows;

The table 4. above can be represented in form of a graph as follows;

Table 5. BIC Gradient Values

Number of Components	BIC Gradient
1	-91254.47116
2	-199467.19126
3	-19703.28905
4	-6610.84811
5	-2619.83574
6	60.39244
7	-23452.38986
8	-3155.35378
9	-1646.44616
10	101.47218
11	-4461.48946
12	-2049.67897
13	369.57305



From the graph above, we see clearly that starting from five components, the gradient becomes almost constant. This means that increasing the number of components does not contribute much gain to the mixture model. Therefore, from this approach, the best model is the one with four components with properties as in table 5

5 Conclusion and Recommendation

Gaussian Mixture Models have been used in Machine Learning to cluster data especially when conducting market research and segmentation of customers. They have also been used in medicine as medical imaging tools, and other areas such as density estimation, anomaly detection, and many more. In this application, these models have been used in fitting financial data exhibiting high-frequency properties, which, cannot be modeled by Gaussian distribution because of fat tails and skewed properties. From this research we find that;

- From section 4.2 we find that the Gaussian Mixtures parameters have been well estimated by use of a powerful machine learning algorithm called Expectation Maximization Algorithm. This algorithm has been applied in this case because in the data we have more than one group of data coming from more than one Gaussian distribution. What is not known is which group of data has come from which Gaussian distribution.

- We have also obtained the curves of the kernel and the fitted distributions and from the look of the different models with different numbers of components, we find that the best model is the one with four components.

- In section 4.3, we computed the BIC scores for each mixture model with an increasing number of components and found out that this criterion fails to penalize over-fitting as the complexity of the mixture models increases. We therefore could not make conclusions on the optimal number of components based on BIC alone.

- As a result, we computed the gradients of BIC (4.3.1) and fiting the curve against the number of components. From this curve, we find that from five components onwards, the BIC gradient is almost constant. This means that from there, increasing the number of components does not contribute any positive change or does not make the model better. From this, we find that the best model is the one with four optimal numbers of components which proves 4.2.3.

We therefore recommend that future studies, especially in the financial market derivatives and high-frequency financial data, apply the concept of Gaussian Mixture modeling in risk management and derivative pricing. We also encourage more research in this field, especially, on the slow convergence of the Gaussian Mixtures to eliminate any possible mistake that might occur as a result of this limitation. The use of Expectation Maximization algorithm in parameter estimation especially in finacial models where there is a possibility of missing information, should be encouraged not only in Gaussian Mixtures but also in other models.
Finally, we encourage the use of Gaussian Mixtures in modeling financial data, unlike the Gaussian distribution which can possibly underestimate risk as a result of its short tails.

References

Cao, J., Li, Z., and Li, J. (2019). Financial time series forecasting model based on ceemdan and lstm. Physica A: Statistical mechanics and its applications, 519:127–139.

Chen, J. (1995). Optimal rate of convergence for finite mixture models. The Annals of Statistics, pages 221–233.

Chen, Y., Georgiou, T. T., and Tannenbaum, A. (2018). Optimal transport for gaussian mixture models. IEEE Access, 7:6269–6278.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. Journal of the royal statistical society: series B (methodological), 39(1):1–22.

Goodhart, C. A. and O'Hara, M. (1997). High frequency data in financial markets: Issues and applications. Journal of Empirical Finance, 4(2-3):73–114.

Goyal, K. and Singhai, J. (2018). Review of background subtraction methods using gaussian mixture model for video surveillance systems. Artificial Intelligence Review, 50:241–259.

Huang, T., Peng, H., and Zhang, K. (2017). Model selection for gaussian mixture models. Statistica Sinica, pages 147–169.

Laird, N., Lange, N., and Stram, D. (1987). Maximum likelihood computations with repeated measures: application of the em algorithm. Journal of the American Statistical Association, 82(397):97–105.

Liao, J. and Berg, A. (2019). Sharpening jensen's inequality. The American Statistician.

Moghar, A. and Hamiche, M. (2020). Stock market prediction using lstm recurrent neural network. Procedia computer science, 170:1168–1173.

Oduor, O. K. and Anyango, C. (2024). Interest rate risk modelling using semi-heavy tail distributions of normal variance-mean mixtures: A study on central bank of kenya interest rates. Asian Journal of Probability and Statistics, 26(11):36–50.

Reddy, A., Ordway-West, M., Lee, M., Dugan, M., Whitney, J., Kahana, R., Ford, B., Muedsam, J., Henslee, A., and Rao, M. (2017). Using gaussian mixture models to detect outliers in seasonal univariate network traffic. In 2017 IEEE Security and Privacy Workshops (SPW), pages 229–234. IEEE.

Sammaknejad, N., Zhao, Y., and Huang, B. (2019). A review of the expectation maximization algorithm in data-driven process identification. Journal of process control, 73:123–136.

Sezer, O. B., Gudelek, M. U., and Ozbayoglu, A. M. (2020). Financial time series forecasting with deep learning: A systematic literature review: 2005–2019. Applied soft computing, 90:106181.

Shahin, I., Nassif, A. B., and Hamsa, S. (2019). Emotion recognition using hybrid gaussian mixture model and deep neural network. IEEE access, 7:26777–26787.

Siami-Namini, S., Tavakoli, N., and Namin, A. S. (2018). A comparison of arima and lstm in forecasting time series. In 2018 17th IEEE international conference on machine learning and applications (ICMLA), pages 1394–1401. Ieee.

Siami-Namini, S., Tavakoli, N., and Namin, A. S. (2019). The performance of lstm and bilstm in forecasting time series. In 2019 IEEE International conference on big data (Big Data), pages 3285–3292. IEEE.

Vijh, M., Chandola, D., Tikkiwal, V. A., and Kumar, A. (2020). Stock closing price prediction using machine learning techniques. Procedia computer science, 167:599–606.

Viroli, C. and McLachlan, G. J. (2019). Deep gaussian mixture models. Statistics and Computing, 29:43–51.

Zong, B., Song, Q., Min, M. R., Cheng, W., Lumezanu, C., Cho, D., and Chen, H. (2018). Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In International conference on learning representations.