INTERNATIONAL JOURNAL OF COMPUTATIONAL MATHEMATICS AND DATA SCIENCE

PATTERN RE-COGNITIVE PERFORMANCE WITH GAUSSIAN MIXTURE MODEL UNI-VARIATE AND MULTIVARIATE PERSPECTIVES

Kalu, S. I.

School of Science & Industrial Technology, Department of Statistics, Abia State Polytechnics, Aba

Corresponding Author: Dr. Kalu, Stephen I, School of Science & Industrial Technology, Department of Statistics, Abia State Polytechnics, Aba

ABSTRACT

This study proposes supervised and unsupervised Algorithm for learning pattern identification performances with finite Gaussian mixture model from uni-variate and multivariate data sets. Bayes' theorem for pattern discrimination was used to relate the probability density function of the data given the class to the posterior probability of the class given the data. The uni-variate case was first considered, with a continuous random variable x, which's pdf, given class C, is a Gaussian with mean μ_c and variance σ_c^2. The study was extended to Multivariate analysis as the number of data dimension $d \geq 2$. The use of finite mixture distributions to control observed and unobserved pattern heterogeneity, has become increasingly popular among those classification dynamic discrete choice models (CDDCMs). One of the challenges of using these discrete choice models is that parameters that could previously be estimated in stages must now be estimated jointly. Actual data discrimination was implemented according to Likelihood-based criterion. The intuition is that since GMM is good at describing actual complicated data, we can model the data using the GMM within one class, then use the PDF of the GMM to calculate the likelihood of any new incoming instances within every class and find the class of which the pdf generates the maximum likelihood. Data sets for both the uni-variate and multivariate cases are in *training and test sets* without missing value(s). The Gaussian discriminant model was trained by about 30% *test sets* of each data sets and the remaining 70% *test sets* was used to test the actual model pattern recognition performances. From the obtained confusion matrix, the overall proportion of test patterns correctly classified by the trained model is about 85%, attributing only 15% to classifications rate otherwise called the apparent error rate. Some vital findings and recommendations were advanced regarding the recognition accuracy of the classifiers using the *training* and *test* data sets. MATLAB Analytical software was used for all data analysis in this study.

Keywords: Finite Mixture, Gaussian Mixture Model, Uni-variate Gaussian Classifier, Multivariate Gaussian Classifier, Classification

Author Biography

Dr. Kalu, Stephen Iheanyichukwu is a Statistics lecturer, in the Department of Statistics, Abia State Polytechnic, Aba, Nigeria. He holds a PhD in Statistics and is currently a Principal Instructor by rank. His specialized realms of scientific research are "Statistical distribution theory mixture modeling optimization and classification in artificial intelligence (AI) via deep machine learning (DML)". Email: monarchynwokike@gmail.com, Orcid number: https://orcid.org/0009-0005-3563-2352.

INTRODUCTION

Finite mixtures are a flexible and powerful probabilistic modeling tool for uni-variate and multivariate data. Mixture models are useful in any area which involves the statistical modeling of data (such as pattern recognition, computer vision, signal and image analysis and machine learning (Nwobi & Kalu, 2023). Statistical pattern recognition, finite mixtures make it possible for a formal probabilistic model-based approach to perform unsupervised learning on a dataset(s) (Titherington et al., 1985; Jain & Dudes, 1988; McLachlan & Basford, 1988; Jain et al., 2000; McLachlan & Peel, 2000). finite mixtures naturally model observations which are assumed to have been produced by one (randomly selected and unknown) set of alternative random source. The usefulness of mixture models is not limited to unsupervised learning applications. Mixture models are able to arbitrarily represent complex probability density functions (pdf). This revealing fact makes mixture model an excellent choice for representing complex class conditional pdf's (i.e., likelihood functions) in Bayesian supervised learning scenarios according to (Kiefer & Wolfowitz, 1956; Streit & Luginbuhl, 1994; Hastie & Tibshirani, 1996; Hinton et al., 1997).

According to Wirjanto & Xu, (2009), a Gaussian Mixture Model (GMM) is a parametric probability density function represented as a weighted sum of Gaussian component densities. GMM are commonly used as a parametric model of the probability of continuous measurements

or features in a biometric system. GMM parameters are estimated from training data using either the iterative Expectation-Maximization (EM) algorithm or Maximum *A Posteriori (MAP)* estimation from a well-trained prior model (Craigmile & Titherington, 1998; Oleszak, 2020). A variable in uni-variate Gaussian classifier just a condition or subset that your data falls into and for example, the analysis might look at a variable of age, height, weight and it doesn't look at more than one variable at a time, otherwise it becomes bi-variate classifier (or in the case of 3 or more, it would be called multivariate classifier (Lindsay, 1995; Hiberty, 1994 & Withers, 1996).

The multivariate Gaussian classifier or joint normal distribution classifier as earlier briefly explained are expansion of the uni-variate normal classifier to higher dimensions D>2 in Probability theory and Statistics. It models the probability in one shot instead of calculating individual probabilities and multiplying them (Kloppenburg & Tavan, 1997; Rangarajan, 1997) for more details. Classification is a machine learning Algorithm that tackles the task of separating distinct component groups and with allocating new observations to previously defined mixture groups (Richard & Dean, 2007)). Classification procedures are less exploratory than discrimination in the sense that they lead to well-defined rules, which can be used for assigning new objects to well defined groups. The major goal of classification is to sort objects (observations) into two or more group categories or more labeled classes. The emphasis of classification is on deriving a rule that can be used to optimally assign new objects to the already labeled classes or categories.

Methods

Uni-variate Gaussian classifier

In this paper, we used Bayes' theorem for pattern recognition (classification), to relate the probability density function of the data given the class to the posterior probability of the class given the data.

Following (Hiroshi, 2015), we first consider the uni-variate case, with a continuous random variable x, whose pdf given class C is a Gaussian with mean μ_i and variance σ_c^2. Using Bayes' theorem, we can write

$$P(C \mid x) \alpha p(X \mid C)P(C)$$

$$\alpha N(x, \mu_i, \sigma_c^2)\, P(C)$$

$$= \frac{1}{\sqrt{2\pi\sigma_c^2}} \exp\left(\frac{-(x-\mu_i)^2}{2\sigma_c^2}\right) p(C) \tag{1}$$

Where $p(x \mid C)$ is the likelihood of class C given an observation x_i.

Log likelihood and log probabilities:

When dealing with Gaussian, it is often important to take logs. We can define the log likelihood of the Gaussian pdf $LL(x \mid C) = LL(x \mid \mu_i, \sigma_c^2)$:

$$LL(x \mid \mu_i, \sigma_c^2) = \ln p(x \mid \mu_i, \sigma_c^2)$$

$$= \ln \frac{1}{\sqrt{2\pi\sigma_c^2}} \exp\left(\frac{-(x-\mu_i)^2}{2\sigma_c^2}\right)$$

$$= -\ln(\sqrt{2\pi\sigma_c^2}) - \frac{(x-\mu_i)^2}{2\sigma_c^2}$$

$$= \frac{1}{2}\left(-\ln(2\pi) - \ln\sigma_c^2 - \frac{(x-\mu_i)^2}{\sigma_c^2}\right) \quad (2)$$

We can use Bayes' theorem to write log posterior probability $LP(C \mid x)$:

$$LP(C \mid x) = LL(x \mid C) + LP(C) + const.$$

$$= \frac{1}{2}\left(-lm(2\pi) - \ln\sigma_c^2 - \frac{(x-\mu_i)^2}{\sigma_c^2}\right) + \ln P(C) + const. \quad (3)$$

Where $LP(C)$ is the log likelihood of class C, and 'const.' is the log of the constant of proportionality, $p(C),$ in equation (1).

Log probability ratio in Bayes' theorem:

If $c_1 \, and \, c_2$ are modeled by Gaussian with means $\mu_1 \, and \, \mu_2$, with variances $\sigma_a^2 \, and \, \sigma_b^2$, then we can write the log odds (ratio of posterior probabilities) as follows:

$$\ln\frac{P(c_1 \mid x)}{P(c_2 \mid x)} = \ln P(c_1 \mid x) - P(c_2 \mid x)$$

$$\ln\frac{P(c_1 \mid x)}{P(c_2 \mid x)} = \frac{1}{2}\left(-\ln(2\pi) - \ln\sigma_a^2 - \frac{(x-\mu_1)^2}{\sigma_a^2}\right) - \frac{1}{2}\left(-\ln(2\pi) - \ln\sigma_b^2 - \frac{(x-\mu_2)^2}{\sigma_b^2}\right) + (\ln P(c_1) - \ln P(c_2))$$

$$= -\frac{1}{2}\left(\frac{(x-\mu_a)^2}{\sigma_a^2} - \frac{(x-\mu_b)^2}{\sigma_b^2} + \ln\sigma_a^2 - \ln\sigma_b^2\right) + \ln P(c_1) - \ln P(c_2). \quad (4)$$

Looking at the right hand side of equation 4, the first two terms are the weighted variance Euclidean distances from the means, the second two terms are log weighted variance (arising from the normalization terms) and the third two terms are the log prior probabilities. Thinking through why intuitively, each of these terms should effect which class one believe best explain an observation pattern.

Multivariate Gaussian Classifier:

Considering D-dimensional data x from class C modeled using a multivariate Gaussian

$$p(x \mid C) = p(x \mid \mu, \Sigma) = N(x; \mu, \Sigma)$$

$$= \frac{1}{(2\pi)^{d/2} |\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right)$$

(5)

The likelihood is given as

$$LL(x \mid \mu, \Sigma) = \ln p(x \mid \mu, \Sigma) = \frac{d}{2}\ln(2\pi) - \frac{1}{2}\ln|\Sigma| - \frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)$$

(6)

and we can write the log posterior probability as

$$\ln p(C \mid x) = -\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu) = \frac{1}{2}\ln|\Sigma| + \ln P(C) + const.$$

(7)

Data presentation, analysis and results

Uni-variate Case

Few data sample points were used to fully demonstrate how the classifier could be applied to pattern recognition/identification

Results

Given a problem with two classes, S and T., some observations (Training data) available in each class are as follows:

Class S	10	8	10	10	11	11
Class T	12	9	15	10	13	13

We assume that each class may be modeled by a Gaussian and that the two classes have equal prior probabilities $P(S) = 0.5,\ P(T) = 0.5$. The maximum likelihood estimators of the mean and variance of each pdf is

$$\hat{\mu}(S) = 10 \qquad \hat{\sigma}(S) = 1$$
$$\hat{\mu}(T) = 12 \qquad \hat{\sigma}(T) = 4$$

The following unlabeled data points are available:

$x_1 = 10, x_2 = 11,\ x_3 = 6$ (Test data)

The problem to resolve now is that to which class should each of the test data points be assigned? This is a two class problem. It is convenient to calculate the log probability ratios for the uni-variate case. (Note: In a multi-class problem, we would calculate the log posterior probability for each class using MATLAB).

Using equations 1-4 on the training data set, we have

$$\ln \frac{P(S \mid X = x)}{P(T \mid X = x)} = -\frac{1}{2}\left(\frac{(x-\mu_s)^2}{\sigma_s^2} - \frac{(x-\mu_T)^2}{\sigma_T^2} + \ln \sigma_S^2 - \ln \sigma_T^2 \right) + \ln P(S) - \ln P(T)$$

$$= -\frac{1}{2}\left(\frac{(x-\mu_S)^2}{\sigma_S^2} - \frac{(x-\mu_T)^2}{\sigma_T^2} + \ln\sigma_S^2 - \ln\sigma_T^2\right)$$

$$= -\frac{1}{2}\left((x-10)^2 - \frac{(x-12)^2}{4} - \ln 4\right)$$

Pattern identification rule and decisions:

If the log ratio is less than 0, assign to T, otherwise assign to S.

$x_1 = 10$

$$\ln\frac{P(S \mid X = x_1)}{P(T \mid X = x_1)} = -\frac{1}{2}\left((x_1-10)^2 - \frac{(x_1-12)^2}{4} - \ln 4\right)$$

$$= -\frac{1}{2}(0-1-\ln 4)$$
$$= 1.19$$

$x_2 = 11$

$$\ln\frac{P(S \mid X = x_2)}{P(T \mid X = x_2)} = -\frac{1}{2}\left((x_2-10)^2 - \frac{(x_1-12)^2}{4} - \ln 4\right)$$

$$= -\frac{1}{2}(1-0.25-\ln 4)$$
$$= 0.32$$

$x_3 = 6$

$$\ln\frac{P(S \mid X = x_3)}{P(T \mid X = x_3)} = -\frac{1}{2}\left((x_3-10)^2 - \frac{(x_3-12)^2}{4} - \ln 4\right)$$

$$= -\frac{1}{2}(16 - 9 - \ln 4)$$
$$= -2.81$$

We assign x_1 to S: x_2 to S and x_3 to T.

Now, assume that the two classes do not have equal prior probabilities. In fact, given that: $P(S) = 0.3,\ P(T) = 0.7$ Using this new prior information, to which class should each of test data points ($(x_1, x_2 . x_3)$ be assigned?

$$\ln \frac{P(S \mid X = x)}{P(T \mid X = x)} = -\frac{1}{2}\left(\frac{(x - \mu_s)^2}{\sigma_s^2} - \frac{(x - \mu_T)^2}{\sigma_T^2} + \ln \sigma_S^2 - \ln \sigma_T^2 \right) + \ln P(S) - \ln P(T)$$

$$= -\frac{1}{2}\left((x - 10)^2 - \frac{(x - 12)^2}{4} + \ln 4 \right) + \ln P(S) - \ln(T)$$

$$= -\frac{1}{2}\left((x - 10)^2 - \frac{(x - 12)^2}{4} - \ln 4 \right) + \ln(3/7)$$

$x_1 = 10$

$$\ln \frac{P(S \mid X = x_1)}{P(T \mid X = x_1)} = -\frac{1}{2}\left((x_1 - 10)^2 - \frac{(x_1 - 12)^2}{4} - \ln 4 \right)$$

$$= -\frac{1}{2}(0 - 1 - \ln 4) + \ln(3/7)$$
$$= 0.34$$

$x_2 = 11$

$$\ln \frac{P(S \mid X = x_2)}{P(T \mid X = x_2)} = -\frac{1}{2}\left((x_2 - 10)^2 - \frac{(x_2 - 12)^2}{4} - \ln 4 \right) + \ln(3/7)$$

$$= -\frac{1}{2}(1 - 0.25 - \ln 4) + \ln(3/7)$$
$$= -0.53$$

$x_3 = 6$

$$\ln \frac{P(S \mid X = x_3)}{P(T \mid X = x_3)} = -\frac{1}{2}\left((x_3 - 10)^2 - \frac{(x_3 - 12)^2}{4} - \ln 4 \right) + \ln(3/7)$$

$$= -\frac{1}{2}(16 - 9 - \ln 4) + \ln(3/7)$$
$$= -3.66$$

We now assign x_1 to S: x_2 to T: x_3 to T.

Multivariate Case

Considering the pattern recognition problem in this study under multivariate case, I used two-dimensional data from three classes assumed to have equal prior probabilities in the case of Multivariate Gaussian classifier. Our training data are stored in the files train A data. Train B.data, Train C.data, with the data in files test A. data.test B.data. and test C.data. One can load the data into MATLAB as follows, then plot the three classes on the same scatter plot (Figure 1) and resolve the pattern recognition problem using the necessary MATLAB syntax source codes.

```
>> xa=trainA;
>> xb=trainB;
```

```
>> xc=trainC.

>> alltest=[testA;testB;testC];


>> % plot the training data figure;

>> hold on;

>> scatter(xa(:,1),xa(:,2),'r','o');

>> scatter(xb(:,1),xb(:,2),'b','x');

>> scatter(xc(:,1),xc(:,2),'c','*');

>>end
```

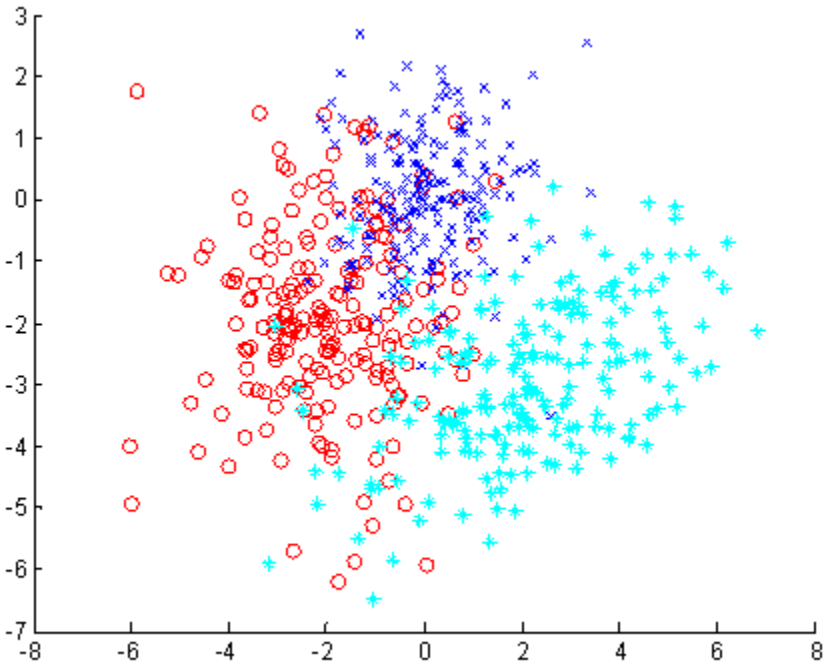


Figure 1: Training data from the classes: A(red circle), B(blue crosses, and C(cyan stars)

There are 200 data points from each class for training, and a further 100 points from each class for testing. The data was generated from Gaussian distributions with the following parameters:

$$\mu_A = \begin{pmatrix} -2 \\ -2 \end{pmatrix} \quad \Sigma_A = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$$

$$\mu_B = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad \Sigma_B = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$\mu_C = \begin{pmatrix} 2 \\ -3 \end{pmatrix} \quad \Sigma_C = \begin{pmatrix} 4.0 & 1.0 \\ 1.0 & 1.5 \end{pmatrix}$$

Each row of the designed matrices xa, xb, xc corresponds to a two –dimensional training data points. To model each class with a Gaussian density, we can immediately estimate the mean and covariance parameters with the sample mean and covariance (computed using built-in MATLAB function *mean* and *cov*):

```
>>% train the Gaussians
>>mua=mean(xa);
>>mub=mean(xb);
>>muc=mean(xc);

>>covara=cov(xa);
>>covarb=cov(xb);
>>covarc=cov(xc);
```

If there are n data points, then cov(x) compute the covariance normalizing by $1/(n-1); \mathrm{cov}(x,1)$ normalizing by $1/n$.

The resulting estimates for the mean and covariance of each class are as follows:

$$\hat{\mu}_A = \begin{pmatrix} -2.1 \\ -2.1 \end{pmatrix} \quad \Sigma_A = \begin{pmatrix} 2.0 & -0.1 \\ -0.1 & -1.6 \end{pmatrix}$$

$$\hat{\mu}_B = \begin{pmatrix} 0.1 \\ 0.1 \end{pmatrix} \quad \Sigma_B = \begin{pmatrix} 1.0 & 0.0 \\ 0.0 & 0.9 \end{pmatrix}$$

$$\hat{\mu}_C = \begin{pmatrix} 2.1 \\ -2.9 \end{pmatrix} \quad \Sigma_C = \begin{pmatrix} 4.0 & 0.8 \\ 0.8 & 1.4 \end{pmatrix}$$

We can now go ahead and classify each testing points. For test points in class A, we do the following using MATLAB software statistical tool;

```
>>testOut=[gauss(mua,covara,testa)  gauss(mub,covarb,testa)  gauss(muc,covarc,testa)];
>>[maxaOut, classa]=max(testaOut,[ ],2);
```

For test points in class B, we have;

The first line of the MATLAB command applies each of the three Gaussian's to all the test points; the second line determines which class has the highest probability. We can repeat this process for each of the training data B and C to see how the test data from each class was classified. Having a look at the result using a confusion matrix corresponds to the predicted classes (i.e classifier output). The two rows correspond to the actual (true) class labels. The number at position (r, c) is the number of patterns from true class r that were classified as class C.

The number of correctly classified patterns is obtained by summing the number on the leading diagonal:

Confusion Matrix Table

Test Data	Predicted class A	B	B
Actual A class	77	15	8
B	5	88	7
C	9	2	89

From this confusion matrix, we can see that the overall proportion of test patterns correctly classified is $(77+88+89/300=245/300=0.85)=85\%.$

Discussion of Results

First, considering the results of our uni-variate analysis, we trained the Gaussian models with a data set coded **Train data**, and the resulting models model was able to accurately classify new in-coming instances to the classes they actually belong after applying the data set coded **Test data** on equations (1)-(7). We explicitly looked at two different assumptions of when: (i) the prior probabilities are the same, and (ii) the prior probabilities are not the same. The classification technique employed in this case is called ***Log Ratio Pattern identification rule***.

In the Multivariate case scenario, we used MATLAB source codes to perform all our analysis which are highly technical and tedious in nature, unlike the uni-variate case, so as for us to achieve highly reliable classification results. The final results of our multivariate analysis are contained in the confusion matrix. As revealed in the confusion matrix Table, the pattern recognition accuracy of the test set is about 85%, hence, attributing only 15% to the error of classifications using Gaussian mixture model.

Conclusion

The classification Algorithms involving Gaussian Mixture Models of two different cases was proposed for this study; the Uni-variate and the Multivariate scenarios, using empirical datasets for complete analytical demonstrations to discriminate component membership. From the results of our analysis, it can be said that our trained Gaussian mixture models performed very well, particularly in the case of the Multivariate analysis, where we measured the classification accuracy rate as shown in the resolved Confusion Matrix Table. The result recorded an impressive correct classification rate of 85% using our trained multivariate model. A study of this kind should be extended to non-Gaussian distributions, like, Exponential distribution, Gamma distribution, Weibull distribution, Binomial distribution, Poisson distribution, Uniform distribution, Bernoulli distribution, Lognormal distribution or a combination of two or more different distributions to see how well the classifier(s) will perform in relation to GMMs as considered in this paper.

References

Craigmile, P.F. & Titherington, D.M. (1998). Parameter Estimationfor finite mixtures of uniform distributions. Communication in Statistics- *Theory and methods,* 26, 1981-1995.

Hastie, T. & Tibshirani, R. (1996). Discriminant analysis by Gaussian Mixture. *Journal of Royal Statl. Society,* 8 (58), 155-176.

Hiberty, C.J. (1994). Applied discriminant analysis. *John Wiley and sons*, New York.

Hinton, G., Dayan, P. & Revow, M. (1997). Modeling for manifolds of images of hand written digits. (8):65-74.

Hiroshi Shimodaira (2015). Classification with Gaussian. Learning and Data Notes, *Informatics2b,* Pg.1-10.

Jain, A.K., Dubes, R. (1988). Algorithm for Clustering Data. Englewood Cliffs, N.J. Prentice Hall.

Kloppenburg, M. & Tavan, P. (1997). Deterministic Annealing for Density Estimation by multivariate normal mixture. *Intl. Statl., Review,* 55, 2089-2092.

Kiefer, J. & Wolfowitz, J. (1956). Consistency of the maximum likelihood estimator in the presence of infinitely many incidental parameters. Annals of mathematical Statistics, 20, 1350-1360.

Lindsay, B.G. (1995). Mixture models: Theory, geometry and applications. Institute for Mathematical Statistics, *Hayward*.

McLachlan, G.J. & Peel, D. (2000). Construction of Gaussian mixture distribution using MATLAB- Infinite mixture models. *Wiley & sons, Inc.*
McLachlan, G.J. & Basford, K.E. (1988). Mixture models: Inference and Applications to clustering, *Marcel Dekker*

Nwobi Felix N., & Kalu Stephen I. (2023). Application of EM Algorithm in classification problem and parameter Estimation of Gaussian Mixture model. *International Journal of Statistics and Applications,* 13(1), 20-26. Doi:10.5923/j.statistics.20231301.03

Olezsak, P. (2020). Estimating the economy with finite mixture models and the EM Algorithm.

Rangarajan A. (1997). Self Annealing: Unifying Deterministic Annealing and relaxation labeling. PP. 229-244.

Richard, A.J., and Dean, W.W. (2007). Applied Multivariate Statistical Analysis, 6th Edition. *Pearson Education Inc.,* PP. 575-595.

Streit, R. & Luginbuhl, T. (1994). Maximum likelihood Training of Probabilistic Neural Networks, (5): 764-783.

Jain, A.K., Duin, R. & Mao, J. (2000). Statistical Pattern Recognition: A Review, 22(1): 4-38.

Titherington, D.M., Smith, A.F.M., & Markov, U.E. (1985). Statistical Analysis of Finite Mixture Distributions. *Wiley and sons Inc.*

Withers, C.S. (1996). Moment Estimates for mixture of several distributions and their different means and scales. *Communication in Statistics, Theory and Methods,* 25,1799-1824.

Wirjanto, T.S., & Xu Dinghai (2009). The Applications of Mixtures of Normal Distributions in Empirical Finance: A selected Survey. *Chinese Economic Society* (CES).