Using Particle Swarm Optimization and Locally-Tuned General Regression Neural Networks with Optimal Completion for Clustering Incomplete Data Using Finite Mixture Models

Ahmed R. Abbas
Using Particle Swarm Optimization and Locally-Tuned General Regression Neural Networks with Optimal Completion for Clustering Incomplete Data Using Finite Mixture Models

Ahmed R. Abas
Department of Computer Science, University College in Lieth, Umm Al-Qura University, Makka Al-Mukarrama, Lieth, Kingdom of Saudi Arabia
armohamed@uqu.edu.sa

Abstract:
In this paper, a new algorithm is presented for unsupervised learning of Finite Mixture Models using incomplete data set. This algorithm applies Particle Swarm Optimization to solve the local optima problem of the Expectation-Maximization algorithm. In addition, the proposed algorithm uses Locally-tuned General Regression neural networks with Optimal Completion Strategy to estimate missing values in the input data set. A comparison study shows the superiority of the proposed algorithm over other algorithms in the literature in unsupervised learning of FMM parameters that produce minimum miss-classification errors when used in clustering incomplete data.

Keywords: Particle Swarm Optimization, Optimal Completion Strategy, Locally-Tuned General Regression Neural Networks, Finite Mixture Models, Unsupervised Learning, Incomplete Data.

1. Introduction

Finite mixture models (FMM) is a semi-parametric method for density estimation and pattern recognition [5] that is used for fitting complex data distributions. This method has the advantage of the analytic simplicity and the advantage of the flexibility to model complex data distributions [11]. Parameters of FMM are usually estimated by the Expectation Maximization (EM) algorithm [9]. The EM algorithm cannot handle incomplete data sets. Therefore, several algorithms are proposed in the literature to modify the EM algorithm to estimate parameters of FMM using incomplete data set [3,16]. Part of these algorithms is affected by the occurrence of outliers in the data, and all of them are affected by the overlap among classes in the data space and the bias in generating the data from its classes [16]. It is shown that better results can be obtained by imputing missing values using the distribution of the input feature vectors rather than using a priori probability distribution function used in the FMM [16,4]. However, these modified EM algorithms have poor performance in learning FMM parameters when clusters of the input data set are largely overlapping and unbalanced in their numbers of feature vectors [16]. This is due to the fact that these algorithms estimate missing values in a certain feature vector only from either parameters or members of one component of the FMM to which this feature vector has the maximum posterior probability. The posterior probability is computed using complete values of each feature vector. This ignores the overlapping among clusters of the data set that is represented by FMM components. This overlapping means that
feature vectors of the data set are generated from different components of FMM with different probabilities. Therefore, these algorithms produce inaccurate estimation of missing values which in turn leads to inaccurate estimation of FMM parameters learned from the whole data set.

In this paper, a new algorithm is proposed to overcome problems of the modified EM algorithms [3,16]. The proposed algorithm is less sensitive to the learning problems of the EM algorithm in cases such as the occurrence of outliers in the data set, the overlapping among data classes, and the unbalanced representation of data classes. The rest of this paper is organized as follows. Section 2 presents the proposed algorithm. Section 3 shows results of the comparison study that is carried out to evaluate the performance of the proposed algorithm. These results are discussed in Section 4. Conclusions are presented in Section 5.

2. The Proposed Algorithm

In the rest of this paper, the modified EM algorithm proposed in [3] is referred to as the MEM algorithm while the modified EM algorithm proposed in [16] is referred to as the LGREM algorithm.

2.1 A Swarm Intelligence Based EM Algorithm

The EM algorithm [9] is sensitive to initialization because it stops at the nearest local maximum to the initial point of the likelihood function [10]. To overcome this problem the proposed algorithm uses Particle Swarm Optimization (PSO) [12, 8] to find the best estimation of FMM parameters that corresponds to the global maximum of the likelihood function. The proposed algorithm uses a swarm of several particles to find the best FMM that fits the input data set. Figure 1 shows the particle structure.

Each particle in the swarm corresponds to a FMM whose parameters are learned by the EM algorithm. After learning, the particle that has the maximum log-likelihood (LOGLH) is selected and its FMM is considered the best model for clustering the input data set.

<table>
<thead>
<tr>
<th>Mixing Weights</th>
<th>Centroids</th>
<th>Covariance Matrixes</th>
<th>LOGLH</th>
</tr>
</thead>
</table>

Figure 1. Particle Structure

All swarm particles are initialized such that their mixing weights are equal and sum to one, centroids are feature vectors chosen randomly from the input data set, covariance matrices are equal to $0.01I_d$, where $I_d$ is the identity matrix of order $d$ and $d$ is the number of features of the data set.

2.2 Locally-Tuned General Regression Neural Networks with Optimal Completion Strategy (OCS)

The OCS [15] is proposed to allow the fuzzy c-means algorithm (FCM) [1] to be used in clustering incomplete data sets [13,14]. In the proposed algorithm, the OCS is used to estimate missing values in the data set using non-parametric estimation values obtained from the locally-tuned general regression neural networks [16] and posterior probabilities of feature vectors containing missing values. This agrees with the basic assumption of clustering using FMM that feature vectors in the data set are generated from different FMM components with different probabilities. The proposed algorithm uses locally-tuned general regression neural networks in estimating missing values [16] because this algorithm produces more accurate estimation of missing values and better learning of FMM.
parameters than using FMM parameters [3] in estimating missing values in the data set [16].

The proposed algorithm uses the PSO-based EM algorithm and the Locally-tuned General Regression neural networks with the OCS to learn FMM parameters from incomplete data. The proposed algorithm is referred to as POLGREM algorithm in the rest of this paper.

2.3 Description of the POLGREM Algorithm

Suppose that the data set \( \mathcal{R} = \{ \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N \} \) consists of \( N \) feature vectors each of which is a vector in \( d \)-feature space such that each feature vector \( \mathbf{x}_i = [x_{i1}, x_{i2}, \ldots, x_{id}]^T \). This data set is assumed to be generated from a FMM of \( K \) multivariate normal distributions with unknown mixing coefficients \( P(c) \), where \( \sum_{c=1}^{K} P(c) = 1 \), and \( 0 \leq P(c) \leq 1 \). Let the probability density of the feature vector \( \mathbf{x}_i \), which is fully observed, given the \( k \)th component in the FMM be \( p(\mathbf{x}_i | \theta_k) = \mathcal{N}(\mathbf{x}_i; \mu_k, \Sigma_k) \), where \( \mu_k \) and \( \Sigma_k \) are the mean and the covariance matrix of this component. The total density of \( \mathbf{x}_i \) from the FMM is then computed as \( p(\mathbf{x}_i) = \sum_{c=1}^{K} P(c) p(\mathbf{x}_i | \theta_c) \). In fitting the FMM, there are two types of missing values that have to be considered; the first type is the values of the cluster membership vector for each feature vector \( \mathbf{z}_i = [z_{i1}, z_{i2}, \ldots, z_{ik}]^T \); the second type is the missing values in the different features of \( \mathcal{R} \). To represent the second type let each feature vector in \( \mathcal{R} \) be rewritten as \( \mathbf{x}_i = (\mathbf{x}_ i^o, \mathbf{x}_i^m) \), where \( o \) and \( m \) superscripts denote the observed and the missing values in this feature vector, respectively.

**Step 1:** Linearly scale the values of each feature in the input data set to make them lie in the interval \([0,1] \). In addition, determine the optimum smoothing parameter \( \sigma \) for each incomplete feature in the data set using the leave-one-out cross validation method. This parameter is used in the locally-tuned general regression neural networks for estimating missing values from neighboring feature vectors in every cluster [16]. The group of fully observed features used in determining \( \sigma \) for a certain feature consists of both the complete features and the features that have smaller missing rates than this feature. The feature vectors that are used in the leave-one-out cross validation method should be observed in the entire fully observed feature group.

**Step 2:** The EM algorithm is initialized several times using a swarm of particles each of which represents a FMM as explained in Section 2.1. The number of particles in the swarm is chosen arbitrary to be twenty in experiments presented in this paper. The FMM corresponding to the maximum log-likelihood function after convergence is selected as the best model for the input data set.

**Step 3:** In the E-step, compute the following quantities for each model component \( c \) in the FMM.

- The posterior probabilities vector \( \mathbf{z}_i \) for all feature vectors in the data set \( \mathcal{R} \).

\[
\hat{z}_i = \frac{\hat{P}(c) p(x_i^o | \theta_c)}{\sum_{j=1}^{K} \hat{P}(j) p(x_i^o | \theta_j)} \tag{1}
\]

- The estimates of the missing values in \( \mathcal{R} \) starting with those values in the feature that has the minimum missing rate. Multiple estimated values are computed for each missing value in the data set (see, Equation 2). Each estimated value is computed from one component in the
FMM using the locally-tuned general regression neural networks [16]. According to the OCS the proposed algorithm estimates the missing value as the average of these multiple estimated values weighted by the posterior probabilities of the feature vector containing the missing value to different components of the FMM (see, Equation 3).

\[ E(\hat{x}_{iq}^c | x_i^q, R) = \frac{\sum_{k=1}^{n_i} x_{ik} \exp(-D_k^2/2\sigma_k^2)r_{kc}}{\sum_{k=1}^{n_i} \exp(-D_k^2/2\sigma_k^2)r_{kc}} \quad (2) \]

\[ E(\hat{x}_{iq} | x_i^q, R) = \sum_{c=1}^{K} z_{ic} E(\hat{x}_{iq}^c | x_i^q, R) \quad (3) \]

where \( R = \{r_{kc}\} \) is the matrix of memberships for each feature vector \( x_i \) to each component \( c \) in the FMM such that \( r_{kc} \) is either one, if \( \tilde{z}_{kc} > \tilde{z}_{kt} \) for all \( t \neq c \), or zero otherwise, \( n_o \) is the number of feature vectors that are observed on both of the observed subspace for the feature vector \( x_i \) and the \( q^{th} \) feature, and \( D_k^2 \) is the squared Euclidean distance between feature vectors \( x_k \) and \( x_i \) on the observed subspace of the feature vector \( x_j \).

After estimating its missing values, the \( q^{th} \) feature is added to the group of fully observed features and this new group is then used in estimating the missing values in the next feature that has the minimum missing rate.

- The necessary statistics for the M-step.

\[ E(z_{iq} | x_i^q, R) = \begin{cases} \tilde{z}_{iq} & x_{iq} \in x_i^q \\ \tilde{z}_{iq} E(x_{iq}^c | x_i^q, R) & x_{iq} \in x_i^m \end{cases} \quad (4) \]

\[ E(z_{iq} | x_i^q, R) = \begin{cases} \tilde{z}_{iq} x_{iq} & x_{iq} \in x_i^q \\ \tilde{z}_{iq} E(x_{iq}^c | x_i^q, R) & x_{iq} \in x_i^m \end{cases} \quad (5) \]

\[ E(z_{iq} | x_i^q, R) = \begin{cases} \tilde{z}_{iq} x_{iq} E(x_{iq}^c | x_i^q, R) & x_{iq} \in x_i^q \\ \tilde{z}_{iq} E(x_{iq}^c | x_i^q, R) E(x_{iq} | x_i^q, R) & x_{iq} \in x_i^m \end{cases} \quad (6) \]

where \( i, q, q' = 1, 2, \ldots, d \), \( E \) is the expectation operator.

**Step 4:** In the M-step, compute parameters of each component \( c \) in the FMM.

\[ \hat{p}(c) = \frac{1}{N} \sum_{j=1}^{N} \tilde{z}_{jc} \quad (6) \]

\[ \hat{\mu}_c = \frac{1}{NP(c)} E(\sum_{j=1}^{N} z_{jc} x_j | x_j^q, R) \quad (7) \]

\[ \hat{\Sigma}_c = \frac{1}{NP(c)} E(\sum_{j=1}^{N} z_{jc} x_j x_j^T | x_j^q, R) - \hat{\mu}_c \hat{\mu}_c^T \quad (8) \]

**Step 5:** After convergence, save the resulting FMM parameters and the total log-likelihood of the feature vectors in the data set in the corresponding particle.

**Step 6:** Repeat Steps 2-5 for every particle in the swarm and then select the best FMM stored in the particle that has the maximum log-likelihood of the data.

**Step 7:** Use the best FMM in estimating missing values in the data set as explained in Step 3 and in clustering feature vectors in the input data set according to Bayes decision rule such that each fully observed feature vector \( x \) is assigned to a certain component \( i \) if \( P(i | x) > P(j | x) \) for all \( j \neq i \), where \( P(i | x) \) is the probability that \( x \) is generated from the component \( i \).

The POLGREM algorithm uses local tuning of the general regression (see, Equation 2) to produce multiple imputations for each missing value in the data set. Each imputation is obtained via a non-linear multivariate regression using a group of fully observed feature vectors belonging to one of the FMM components. These groups are
obtained using Bayes decision rule. Then, the POLGREM estimates the missing value as the average of these multiple imputations weighted by the posterior probabilities of the feature vector containing the missing value to different components of the FMM. This makes the POLGREM algorithm be insensitive to global correlations between features of the input data set. Therefore, the POLGREM does not have the limitation of the general regression neural networks [2] when used for estimating missing values without local tuning with weakly correlated data [16]. In addition, local tuning of the general regression neural networks used in the POLGREM algorithm makes the algorithm be less dependent of FMM parameters. Therefore, the POLGREM algorithm overcomes the limitation of the MEM algorithm with small data sets which may contain outliers, overlapping clusters, or large differences in the sizes of their clusters [16]. Finally, due to the use of the OCS in estimating missing values the POLGREM algorithm produces accurate estimation of both the missing values in the data set and FMM parameters, especially when clusters of the data set are largely overlapping and different in their sizes. Therefore, the POLGREM algorithm overcomes problems of cluster overlapping and unbalanced cluster sizes that are limitations of the algorithm proposed in [16], which is referred to as the LGREM algorithm in the experiments presented in this paper.

3. Experiments and Results

The POLGREM algorithm is evaluated and compared with a number of algorithms proposed in the literature in learning FMM parameters for clustering incomplete data sets. These algorithms are the LGREM algorithm [16] and the MEM algorithm [3]. The comparison study in this paper uses two real data sets. The missing values are randomly placed with different missing rates in two features of each data set. These features are selected such that the visual separation among data classes is maximum. The mechanism of the occurrence of the missing values in all data sets is missing completely at random (MCAR) [7]. These data sets are described as follows.

**The Iris Dataset:**

The Iris data set [6] contains 150 feature vectors each of which is a vector in four-feature space. These feature vectors represent three classes each of which has 50 feature vectors belonging to it. Two of these classes are overlapping in the data space. Correlations between different pairs of features of this data set are too weak. The missing values are put in the third and in the fourth data features. In addition, each one of the FMMs learnt by all algorithms compared in this study consists of three Gaussian components that have non-restricted covariance matrices.

**The Pima Indians Diabetes Dataset:**

The Pima Indians Diabetes data set contains 768 feature vectors each of which is a vector in eight-feature space. These feature vectors represent two classes; the first class has 500 feature vectors belonging to it; the second class has 268 feature vectors belonging to it. These classes are largely overlapping. Correlations between different pairs of features of this data set are too weak. The missing values are put in the second and in the fifth features of the data set. Each one of the FMMs learnt by all algorithms compared in this study consists of two Gaussian components that have non-restricted covariance matrices.

The evaluation criterion used in this study to compare the different algorithms is

---

1 The Iris and the Pima data sets are available at: http://archive.ics.uci.edu/ml/datasets.html
the Miss-Classification Error (MCE). It is computed by comparing the clustering results, obtained using Bayes decision rule, of the learned FMM with the true classification of the data feature vectors, assuming each class is represented by a component in the FMM. Components of the FMM are allocated to different data classes such that the total number of misclassified feature vectors in the data set is minimum. Let the number of feature vectors belonging to class \( i \) be \( N_i \), from which \( N_i^m \) feature vectors are not clustered into the component that represents this class in the FMM. Then the MCE for class \( i \) is computed as

\[
MCE_{\text{class}_i} = \frac{N_i^m}{N_i}.
\]

Assuming the data set is generated from \( K \) classes the total MCE is the average of all the class-MCEs and it is computed as

\[
MCE_T = \frac{1}{K} \left( \sum_{i=1}^{K} MCE_{\text{class}_i} \right).
\]

Tables 1 and 3 show comparisons of different pairs of the algorithms using the Student’s paired t-test statistic with each one of the data sets. The P-value is the significance and the T-value is the t-statistic. This test examines the statistical significance of the difference in performance of pairs of algorithms using their total MCEs obtained from ten different experiments. In each experiment, a different group of feature vectors is randomly selected to contain missing values. The results of this test are shown for each pair of percentages of missing values in two different features. The shaded cells in each table represent the cases in which the difference in performance of certain pairs of algorithms is statistically significant according to the 5% significance level. Tables 2 and 4 show comparisons of the algorithms using the mean (MCE) and the standard deviation (STD) of the total Miss-Classification Error obtained from ten different experiments using each one of the data sets. The shaded cells in each table represent the minimum value of the MCE among all algorithms compared.

4. Discussion of Results

Tables 1 and 2 show that the performance of the POLGREM algorithm is not superior (\( T > 0 \) in most cases) and the LGREM algorithm has the best performance among all algorithms compared (MCE is the minimum). These results show that the performance of the POLGREM algorithm is not superior when the relations among clusters of the input data set in the whole space are not maintained in the subspace that is composed of the complete features. For example, the Iris data set has clusters that are partially overlapping in the whole feature space but largely overlapping in the complete feature subspace. This change in relations among clusters happens when the complete subspace of the input data set does not contain enough information to represent clusters of the input data. This results in inaccurate computation of the posterior probabilities and estimation of the missing values. In this case, the LGREM is superior because it is less depending on the posterior probabilities in estimating the missing values.

Tables 3 and 4 show that the performance of the POLGREM algorithm is significantly different from all other algorithms (\( P \leq 0.05 \)) and better than them (\( T < 0 \) and MCE is the minimum). These results show that the performance of the POLGREM algorithm is superior when the input data set is generated from largely overlapping clusters and these clusters are of different sizes. Although the Pima data set contains weakly correlated features and largely overlapping clusters of different sizes the POLGREM has the best performance among all other algorithms compared. This is because clusters of this data set are largely overlapping in the whole feature space and in the subspace that is composed of complete features.

In general, the performance of the POLGREM algorithm proves superiority over
the other algorithms compared in this paper when the input data set is generated from overlapping clusters that are of different sizes (see the results with the Pima data set in Tables 3 and 4). This is due to the use of the OCS and the locally-tuned general regression neural network in the POLGREM algorithm that considers the contribution of different clusters in the data set in estimating the missing values which results in better estimation of both missing values in the data set and FMM parameters. On the other hand, the performance of the POLGREM is not superior when relations among clusters of the data in the whole feature space are not maintained in the subspace that is composed of the complete features (see the results with the Iris data sets in Tables 1 and 2). This is due to the use of posterior probabilities in estimating missing values in the data set and these probabilities are computed using the complete features of the data set. Therefore, if the subspace that is composed of the complete features of the data set does not maintain relations of clusters of the data in the whole feature space the computation of the posterior probabilities is not accurate. This in turn leads to inaccurate estimation of both missing values in the data set and FMM parameters.

5. Conclusions

In this paper, the POLGREM algorithm is proposed to overcome problems of algorithms used in the literature in learning FMM parameters for clustering using incomplete data sets. A comparison study shows the superiority of the POLGREM algorithm over other algorithms compared when the input data set may contain clusters that are largely overlapping, or clusters that have large differences in their sizes. To be superior in its performance the POLGREM algorithm requires the relations among clusters of the data set in the whole feature space to be maintained in the subspace that is composed of complete features of the data set. In the future, the POLGREM should be developed to be less dependent on posterior probabilities to enhance its performance when cluster relations change in the complete subspace of the data set.

References:


Table 1. Comparison Results of the Student's Paired T-Test Statistic with the Iris Data Set.

<table>
<thead>
<tr>
<th>Missing %</th>
<th>T(POLGREM ,LGREM)</th>
<th>T(POLGREM ,MEM)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P</td>
<td>T</td>
</tr>
<tr>
<td>(5%,10%)</td>
<td>0.69</td>
<td>0.41</td>
</tr>
<tr>
<td>(10%,20%)</td>
<td>0.91</td>
<td>0.11</td>
</tr>
<tr>
<td>(15%,30%)</td>
<td>0.59</td>
<td>0.56</td>
</tr>
<tr>
<td>(20%,40%)</td>
<td>0.01</td>
<td>3.04</td>
</tr>
<tr>
<td>(25%,50%)</td>
<td>0.02</td>
<td>2.94</td>
</tr>
</tbody>
</table>

Table 2. Comparison Results of the Mean and the Standard Deviation of the Total Mis-Classification Error with the Iris Dataset.

<table>
<thead>
<tr>
<th>Missing %</th>
<th>POLGREM</th>
<th>MEM</th>
<th>LGREM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MCE</td>
<td>STD</td>
<td>MCE</td>
</tr>
<tr>
<td>(5%,10%)</td>
<td>0.141</td>
<td>0.169</td>
<td>0.349</td>
</tr>
<tr>
<td>(10%,20%)</td>
<td>0.091</td>
<td>0.093</td>
<td>0.361</td>
</tr>
<tr>
<td>(15%,30%)</td>
<td>0.169</td>
<td>0.124</td>
<td>0.375</td>
</tr>
<tr>
<td>(20%,40%)</td>
<td>0.258</td>
<td>0.101</td>
<td>0.321</td>
</tr>
<tr>
<td>(25%,50%)</td>
<td>0.249</td>
<td>0.095</td>
<td>0.353</td>
</tr>
</tbody>
</table>

Table 3. Comparison Results of the Student's Paired T-Test Statistic with the Pima Dataset.

<table>
<thead>
<tr>
<th>Missing %</th>
<th>T(POLGREM ,LGREM)</th>
<th>T(POLGREM ,MEM)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P</td>
<td>T</td>
</tr>
<tr>
<td>(5%,10%)</td>
<td>0.00</td>
<td>-38.03</td>
</tr>
<tr>
<td>(10%,20%)</td>
<td>0.00</td>
<td>-27.46</td>
</tr>
<tr>
<td>(15%,30%)</td>
<td>0.00</td>
<td>-23.82</td>
</tr>
<tr>
<td>(20%,40%)</td>
<td>0.00</td>
<td>-16.19</td>
</tr>
<tr>
<td>(25%,50%)</td>
<td>0.00</td>
<td>-34.25</td>
</tr>
</tbody>
</table>

Table 4. Comparison Results of the Mean and the Standard Deviation of the Total Mis-Classification Error with the Pima Dataset.

<table>
<thead>
<tr>
<th>Missing %</th>
<th>POLGREM</th>
<th>MEM</th>
<th>LGREM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MCE</td>
<td>STD</td>
<td>MCE</td>
</tr>
<tr>
<td>(5%,10%)</td>
<td>0.037</td>
<td>0.009</td>
<td>0.440</td>
</tr>
<tr>
<td>(10%,20%)</td>
<td>0.039</td>
<td>0.012</td>
<td>0.393</td>
</tr>
<tr>
<td>(15%,30%)</td>
<td>0.033</td>
<td>0.007</td>
<td>0.394</td>
</tr>
<tr>
<td>(20%,40%)</td>
<td>0.036</td>
<td>0.010</td>
<td>0.356</td>
</tr>
<tr>
<td>(25%,50%)</td>
<td>0.041</td>
<td>0.009</td>
<td>0.367</td>
</tr>
</tbody>
</table>