

[Egyptian Informatics Journal (2010) 11, 49](http://dx.doi.org/10.1016/j.eij.2010.10.001)–[57](http://dx.doi.org/10.1016/j.eij.2010.10.001)

Cairo University

Egyptian Informatics Journal

[www.elsevier.com/locate/eij](http://www.elsevier.com/locate/eij) [www.sciencedirect.com](http://www.sciencedirect.com/science/journal/11108665)

ORIGINAL ARTICLE

Using general regression with local tuning for learning mixture models from incomplete data sets

Ahmed R. Abas

*Department of Computer Science, Faculty of Computers and Informatics, Zagazig University, Zagazig, Egypt*

Received 4 May 2010; accepted 24 June 2010

Available online 2 November 2010

Abstract Finite mixture models is a pattern recognition technique that is used for fitting complex data distributions. Parameters of this mixture models are usually determined via the Expectation Maximization (EM) algorithm. A modified version of the EM algorithm is proposed earlier to han- dle data sets with missing values. This algorithm is affected by the occurrence of outliers in the data, the overlap among classes in the data space and the bias in generating the data from its classes. In addition, it only works well when the missing value rate is low. In this paper, a new algorithm is proposed to overcome these problems. A comparison study shows the superiority of the new algo- rithm over the modified EM algorithm and other algorithms commonly used in the literature.

© 2010 Faculty of Computers and Information, Cairo University.

KEYWORDS

Finite mixture models; Expectation Maximization; Missing value;

Incomplete data set; Locally tuned general regression

Production and hosting by Elsevier B.V. All rights reserved.

1. Introduction

Finite mixture models (FMM) is a semi-parametric method for density estimation and pattern recognition [[1]](#_bookmark16). It has the advan- tage of the analytic simplicity of the parametric methods and the advantage of the flexibility to model complex data distribu-

tions of the non-parametric methods [[2]](#_bookmark17). Parameters of FMM are usually estimated by the Expectation Maximization (EM) algorithm [[3]](#_bookmark18). It is shown that the EM algorithm can estimate parameters of a multivariate normal distribution from a data set with missing values [[4]](#_bookmark22). The EM algorithm is modified to estimate parameters of a mixture of multivariate normal distri-

butions using incomplete data set [[5]](#_bookmark24). The modified EM algo-

E-mail address: [arabas@zu.edu.eg](mailto:arabas@zu.edu.eg)

1110-8665 © 2010 Faculty of Computers and Information, Cairo University. Production and hosting by Elsevier B.V. All rights reserved.

Peer review under responsibility of Faculty of Computers and Information, Cairo University.

doi:[10.1016/j.eij.2010.10.001](http://dx.doi.org/10.1016/j.eij.2010.10.001)

Production and hosting by Elsevier

rithm is used in clustering data sets that contain missing values and a number of categorical features [[6,7]](#_bookmark12). Also, it is used in learning parameters of the radial basis function networks used in classifying data sets that contain missing values [[8]](#_bookmark12). When there is sufficiently large number of observed values, this algorithm outperforms the EM algorithm combined with either the unconditional mean imputation [[9,10]](#_bookmark12), or the conditional mean imputation [[11]](#_bookmark12), of the missing values according to the classification performance of the resulting FMM. Also, this algorithm outperforms the EM algorithm combined with delet- ing feature vectors that contain missing values [[11]](#_bookmark12), according to the classification performance of the resulting FMM.

However, the outcome of the modified EM algorithm is influenced by several modelling assumptions such as the num- ber of components and the probability distribution function of each component [[11,12]](#_bookmark12). For example, the modified EM algo- rithm produces an inaccurate estimation of both FMM param- eters and missing values in the input data set when initialised with an FMM that has too few components, each of which has a probability distribution function that does not fit well any of the clusters of the input data set. In addition, this algo- rithm has a poor performance when the size of the data set is small [[11]](#_bookmark12). It is shown that better results can be obtained by imputing missing values using the distribution of the input fea- ture vectors rather than using a priori probability distribution function used in the FMM [[11]](#_bookmark12). The modified EM algorithm

[[5]](#_bookmark24) is slightly changed to reduce computational burden during the EM iterations by incorporating two types of auxiliary bin- ary indicator matrices corresponding to the observed and unobserved components of each datum [[13]](#_bookmark12). The resulting EM algorithm is compared with a novel Data Augmentation (DA) computational algorithm for learning normal mixture

models when the data are missing at random [[13]](#_bookmark12). Experimen-

fore there is no assurance that the error in estimating the regression value for any new point will be small. Similar algo- rithms to the GRNN are used to train neural networks, in which the hidden neurons use Gaussian density functions, using data sets that contain missing values [[16,17]](#_bookmark15). These algo- rithms are used for regression and classification and they exhi- bit better performances than neural networks that are trained using either only the complete portion of the input data set or the total data set after estimating its missing values using the unconditional mean imputation method.

Suppose that R = {x1; x2; ... ; xN} is a data set that con-

tains N feature vectors each of which is a vector in d-feature space such that x = [*x* ; *x* ; .. . ; *x* ; *y* ]*T*, where the first (d- 1)-features are complete and the d-th feature contains missing

*i i*1 *i*2 *i*(d-1) *i*

values. The GRNN algorithm imputes the missing value in x*i* by computing the weighted average of all the observed values on the d-th feature such that each observed value is weighted exponentially according to its Euclidean distance from x*i* in the fully observed subspace.

2 Pn*o y* exp *D*

*j*

*y*^ (x )=

2r2

*o*

*j*=1

exp — *j*

—

*j*=1 *j*

2r2

(1)

tal results show that DA imputation exhibits considerable

promising accuracy in the prediction of missing values when

compared to the EM imputation, especially as the missing rate

*i i* Pn *D*2

increases. However, both algorithms impute missing values

where *no* is the number of feature vectors that are fully

observed, *D*2 = Pd—1(x — x )2 is the Euclidean distance

using mixture model parameters and hence their imputations

*j*

*h*=1

*ih*

*jh*

are sensitive to the prior information about density functions of mixture components and the size of data that are fully ob- served. A supervised classification method, called robust mix- ture discriminant analysis (RMDA), is proposed to deal with label noised data [[14]](#_bookmark13). The proposed algorithm uses only fully observed data to learn mixture model parameters and then uses this mixture model to estimate labels and detect noisy ones. However, imputations made by this algorithm are sensi- tive to prior information about density functions of mixture components, the size of data that are fully observed and assumptions such as all the uncertain labels are in one feature. In this paper, a new algorithm is proposed to overcome problems of the modified EM algorithm [[5,13,14]](#_bookmark24). The pro- posed algorithm is less sensitive to the modelling assumptions than the modified EM algorithm in estimating missing values. In addition, it is less affected by learning problems than the EM algorithm in cases such as the occurrence of outliers in

the data set and the overlap among data classes.

1. The general regression neural network

The general regression neural network (referred to as GRNN in the rest of this paper) implements a non-parametric algo- rithm for density estimation and general (non-linear) regres- sion [[15]](#_bookmark14). This algorithm has a better regression performance than conventional non-linear regression techniques, particu- larly with a sparse data set that has a small number of feature vectors and a large number of features. This is because the regression surface resulting from the GRNN is defined every- where in the data space [[15]](#_bookmark14). For example, when the input data set is sparsely distributed, the smoothing parameter (window size) used in the GRNN can be large in order to estimate the regression value at any new point with similar accuracy to the other points. On the other hand, conventional non-linear regression techniques tend to overfit this data set, and there-

between x*i* and x*j*, and r is the smoothing parameter for the d-th feature. To identify the optimum value of r, the GRNN uses the leave-one-out cross validation method. The optimum r is then used in imputing all the missing values on the d-th feature.

However, as a non-linear regression technique the GRNN algorithm produces good results when the data set has a large number of features that are strongly correlated [[18–20]](#_bookmark19). Also, it requires the number of missing values to be small to accelerate the algorithm and to produce unbiased imputations [[18,4,15,20].](#_bookmark19) Therefore, this algorithm is inappropriate for a small data set, which has a small number of feature vectors and a small number of features, especially when several fea- tures contain missing values and the correlations between these features and the other features in the data set are weak.

In this paper, the GRNN algorithm is used to fill in the missing values in a certain data set to be used by the EM algo- rithm for learning parameters of the FMM. The resulting algo- rithm is referred to as the GREM algorithm in the rest of this paper and it is used in the comparison study below to investi- gate the performance of the proposed algorithm against other algorithms.

1. The modified EM algorithm

The EM algorithm is modified to estimate FMM parameters from a data set with missing values [[5]](#_bookmark24). This algorithm is referred to as the MEM algorithm in this paper. In this algorithm, the missing values in the input data set and some other statistics are estimated in the E-step from the parameters of each model component. These values together with the observed values in the data set are then used in the M-step to estimate the FMM parameters. This means that in the E-step multiple imputations are obtained for each missing va- lue in the input data set from the different components in the FMM and then these imputations are used in the M-step to

estimate the FMM parameters. The following paragraphs give a more detailed explanation of the MEM algorithm.

N

1 X

*P*b(*c*)= N

*j*=1

*z*^*jc* (8)

1 2 N !

X

Suppose that the data set R = {x ; x ; .. . ; x

} consists of N

feature vectors each of which is a vector in d-feature space such that each feature vector x*i* = [*xi*1; *xi*2; ... ; *xi*d]*T*. This data set is assumed to be generated from a FMM of K multivariate nor-

b

*j*

mal distributions with unknown mixing coefficients *P*(*c*), where

l^*c* =

1

N*P*(*c*)

1

N

*E*

*j*=1

XN

*zjc*x*j*|x*o*; h*c*

!

(9)

PK R^*c* =

*E zjc*x*j*x*T*|x*o*; h*c*

— l^*c*l^*T*

(10)

*c*=1

*P*(*c*)= 1, and 0 6 *P*(*c*) 6 1. Let the probability density of

N*P*b(*c*)

*j j c*

*j*=1

the feature vector x*i*, which is fully observed, given the *k*th com- ponent in the FMM be *p*(x*i*|h*k*)= *N*(x*i*; l*k*; R*k*), where l*k* and R*k* are the mean and the covariance matrix of this component.

The total density of x*i* from the FMM is then computed as *p*(x )= K *P*(*c*)*p*(x |h ). 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

*i*

*c*=1

*i*

*c*

P

vector z = [*z* ; *z* ; ... ; *z* ]*T*; the second type is the missing val-

Both of the E-step and the M-step are repeated until the algo- rithm converges to a local maximum of the likelihood function.

1. Comparing the MEM algorithm with the GREM algorithm in unsupervised learning of FMM parameters

*i i*1 *i*2 *i*K

ues in the different features of R. To represent the second type

let each feature vector in R be rewritten as x*i* = (x*o*; x*m*), where

The same notation that is introduced in Section 3 is used in this

*i i* section. To simplify the comparison, it is assumed that the data

*o* and *m* superscripts denote the observed and the missing val-

ues in this feature vector, respectively.

In the E-step of the MEM algorithm, all z*i*s are approxi- mated by the posterior probabilities using the observed values for each feature vector x*i* and the parameters of each compo- nent in the FMM as follows.

*P*b(*c*)*p*(x*o*|h*c*)

*z*^

= *i*

(2)

set R is generated from a mixture model whose components are compact and well separated in the data space. Therefore, each feature vector in the data set belongs to only one compo- nent in the FMM to which its posterior probability is 1. In

addition, it is assumed that each component in the FMM that needs to be learned from the data set R has a spherical Gauss- ian distribution. Finally, it is assumed that there is sufficiently

*ic* PK

*j*=1

*i*

*j*

*P*b(*j*)*p*(x |h )

large number of observed values in the data set R to learn the

FMM parameters accurately.

Let the mean and the covariance matrix for each component in

the FMM be partitioned in the same manner as the feature vector x*i* with *o* and *m* superscripts denoting the components corresponding to observed and missing features for this feature

In the MEM algorithm, the missing values in the feature

vector x*i* are imputed, given the parameters of the component *c* in the FMM, as shown in Eq. [(4)](#_bookmark2). Since all components in the FMM have spherical shapes then all the values in R*mo* are

*c*

vector.

l*o*

l =

; R =

R*oo* R*om*

zeros. Therefore, Eq. [(4)](#_bookmark2) can be rewritten as follows.

*E*ÿx*m*|x*o*; h = l*m*

*i*

*i*

*c*

*c*

(11)

l*m*

R*mo* R*mm*

Then the expectation of the missing values in the feature vector x*i* given the parameters of the component *c* is computed as follows.

(3)

Since the posterior probability of each feature vector in R is 1 for precisely one component in the FMM then Eq. (9) can be rewritten as follows.

1 XNc !

*m*

*m o*

*E*(x*m*|x*o*; h )= l*m* + R*mo*R*oo*—1 (x — l*o*) (4)

l*c* = N *E*

x*j* |x*j* ; h*c*

(12)

*i i c*

*c c c i c*

*c j*=1

Also, the residual covariances of the estimated values for this feature vector are computed as follows.

where N*c* is the number of feature vectors that belong to the component *c*.

*m mT o*

*mm mo*

*oo*—1

*moT*

On the other hand, the GREM algorithm imputes each

Cov(x*i* x*i* |x*i* ; h*c*)= R*c* — R*c* R*c* R*c* (5) \_ *o*

Also, some statistical moments necessary for each model com- ponent in the M-step are computed as follows.

missing value *xiq*(x*i* ) in the feature vector x*i* according to

Eq. [(1)](#_bookmark0). This equation can be rewritten as follows:

PK P*n*h *x*

*z*^*ic xij xij* ∈ x*o*

*h*=1

*o*

*j*=1

exp — *D*2

*E*(*zicxij*|x*i* ; h*c*)=

*o*

*ic*

*ij*

*c*

*ij*

*o*

*j*=1

*z*^ *E*(*x* |x*o*; h ) *x*

*i*

∈ x*m*

*i*

*i*

(6)

*iq i*

PK P*n*h

*h*=1

*q*

(13)

*D*2

2r2

*jq*

*j*

2r2

exp — *j*

*q*

8> *z*^*icxijxij*' *xij* ; *xij*' ∈ x*o*

*x*^ (x*o*)=

>

*o*

*m*

*i*

where *nh* is the number of the feature vectors that belong to the

*z*^ *x E*(*x* ' |x ; h ) *x* ' ∈ x

>< *ic ij ij i c ij* *i*

*o*

*o*

*m*

the fully observed subspace for the feature vector x*i* and the

*E*(*zicxijxij*' |x*i* ; h*c*)=

*z*^*icE*(*xij*|x*i* ; h*c*)*xij*' *xij* ∈ x*i*

*j*

*o*

component *h* in the FMM, and they are fully observed on both

*q*th feature. *D*2

is the squared Euclidean distance on the

*o*

*o*

>

*z*^*ic*[*E*(*xij*|x*i* ; h*c*)*E*(*xij*' |x*i* ; h*c*)

>:

+Cov(*xij*; *xij*' |x*o*; h*c*)] *xij*; *xij*' ∈ x*m*

the feature vector x*j* that belongs to component *h*.

observed subspace of the x*i* between this feature vector and

*i*

where *i*; *j*; *j*' = 1; 2; ... ; d.

*i*

(7)

Let the correlations between different pairs of features of R be strong such that feature vectors that belong to one cluster are near to each other in all subspaces of the data space. Since

In the M-step of the MEM algorithm, the new estimates of

the FMM parameters are computed as follows.

R is generated from a mixture of compact and well-separated

clusters in the data space then, in all subspaces of R the

distances between feature vectors that belong to the same clus- ter are small compared with the distances between feature vec- tors that belong to different clusters. Therefore, the exponential weight in Eq. [(13)](#_bookmark3) of a certain feature vector x*j* is approximately one, if x*j* and x*i* belong to the same cluster, and is zero otherwise. Then Eq. [(13)](#_bookmark3) can be rewritten as follows.

1 *o*

n*c*

X

*x*^*iq*(x*o*)ﬃ *xjq* = l^*cq* (14)

*i*

*n*

*c*

*o j*=1

General Regression with the Expectation Maximization (LGREM) algorithm.

*5.1. Description of the LGREM algorithm*

In this section, the same notation that is introduced in Section 3 is used.

*Step 1:* Linearly scale the values of each feature in the input data set to make them lie in the interval [0,1]. This process removes the effect of different unit scales on different fea-

Eq. [(14)](#_bookmark4) is then generalized as follows.

x^*m*(x*o*)ﬃ l^*m*

(15)

tures, and therefore it is essential for finding the true cluster structure of this data set and the contribution of each fea-

*i i c*

Eqs. [(11) and (15)](#_bookmark1) show that the GREM algorithm and the MEM algorithm are similar in their estimation of the missing

values when there are strong correlations between different pairs of features of R.

Let the correlations between different pairs of features of R

be weak such that feature vectors that belong to different clus- ters can be near to each other in all subspaces of R. Then, the estimation of the missing values in the GREM algorithm, which is produced by Eq. [(13)](#_bookmark3), is approximately the average

of all the observed values on the *q*th feature. This affects the learning of the FMM parameters such that the resulting com- ponents will be overlapping in the whole space. Therefore using this mixture model for clustering cannot produce com- pact or well-separated clusters. On the other hand, the estima- tion of the missing values in the MEM algorithm, which is produced by Eq. [(11)](#_bookmark1), results in different imputations for each missing value from the parameters of the different components in the FMM. When used in estimating parameters of each component in the FMM, these different imputations preserve the features of these components that are learned from the

fully observed values in R. Therefore, using the resulting

FMM for clustering produces compact and well-separated clusters. In general, when the correlations between different pairs of features of R are weak the MEM algorithm outper-

ture in the creation of this structure. A comparison of sev-

eral methods of data normalization for cluster analysis has shown the superiority of the linear scaling method over many other normalization methods before applying cluster- ing algorithms in terms of the resulting cluster separation and error condition [[22,23]](#_bookmark20). In addition, determine the opti- mum smoothing parameter r for each incomplete feature in the data set using the leave-one-out cross validation method. The group of fully observed features used in deter- mining r 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:* As the EM algorithm converges toward the nearest local maximum of the likelihood function to the starting point [[24]](#_bookmark21), initialise the EM algorithm randomly several times (20 times in our experiments) and select the FMM corresponding to the maximum log-likelihood function after convergence 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 z*i* for all feature vectors in the data set R.

*P*b(*c*)*p*(x*o*|h*c*)

*z*^

= *i*

(16)

*j*=1

*i*

*j*

forms the GREM algorithm with respect to the clustering per-

formance of the resulting FMM.

*ic* PK

*P*b(*j*)*p*(x*o*|h )

1. The proposed algorithm for learning FMM parameters

It is argued in the previous section that the MEM algorithm is

* + The estimates of the missing values in R starting with those values in the feature that has the minimum missing rate.

less sensitive to the correlations between different pairs of fea-

tures of the data set than the GREM algorithm. However, the

*k*=1

*kq*

2r2

*kc*

2

P*n*0 *x*

exp *D*2 *r*

performance of the MEM algorithm in estimating FMM

*q*

\_*c*

0

parameters is severely distorted when the size of the input data

*E xiq* |x*i* ; R =

2r2

(17)

set is small [[11,21]](#_bookmark12). For example, when the input data set has a small size and it contains a number of outlier feature vectors,

0

*k*=1

exp *D*

*q*

*rkc*

P*n*

where R = {*rkc*} is the matrix of memberships for each fea- ture vector x*k* to each component *c* in the FMM such that *rkc*

\_ \_

an overlap among its clusters, or a large difference in the sizes of its clusters the estimates produced by the MEM algorithm for the FMM parameters are largely biased and inaccurate.

In this section, a new algorithm is proposed to overcome problems of both the GREM algorithm and the MEM algo- rithm when dealing with a small incomplete data set that may contain outliers, overlapped clusters, or a large difference in the sizes of its clusters. These problems are the sensitivity to global correlations between features of the input data set, and the dependence on FMM parameters learned from this data set. The proposed algorithm is referred to as the Locally-tuned

is either one, if *z kc z kt* for all *t* – *c*, or zero otherwise, *no* 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 fea- ture, and *D*2 is the squared Euclidean distance between fea- ture vectors x*k* and x*i* on the observed subspace of the feature vector x*i*.

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.

*k*

* The necessary statistics for the M-step.

*E*(*zicxiq*|x*o*; R)=

*i*

*z*^*ic xiq xiq* ∈ x*o*

*z*^ *E*(*xc* |x*o*; R) *x* ∈ x*m*

*i*

(

(18)

it helps the algorithm to be less dependent of FMM param- eters, and therefore to overcome the limitation of the MEM

*ic iq i*

*iq* *i*

algorithm (see Eq. [(4)](#_bookmark2)) with small data sets, which may con- tain outliers, overlapped clusters, or large differences in the sizes of their clusters.

*i*

>><

*z*^*icxiqE*(*xc* ' |x*o*; R) *xiq*' ∈ x*m*

*o*

*E*(*zicxiqxiq*' |x*i* ; R)=

*iq*

*o*

*i*

*i*

*m*

*z*^*icE*(*xiq* |x*i* ; R)*xiq*' *xiq* ∈ x*i*

*c*

>>:*z*^*icE*(*xc* |x*o*; R)*E*(*xc* ' |x*o*; R) *xiq*; *xiq*' ∈ x*m*

8>*z*^*ic xiqxiq*' *xiq* ; *xiq*' ∈ x*o*

1. Experiments and results

The LGREM algorithm is evaluated and compared with a number of commonly used algorithms in the literature in unsu-

*iq i*

*iq i*

*i*

(19)

pervised learning of FMM parameters using data sets with missing values. These algorithms are the MEM algorithm,

where *i*; *q*; *q*' = 1; 2; .. . ; d, *E* is the expectation operator.

*Step 4:* In the M-step, compute parameters of each compo- nent *c* in the FMM.

1 XN

X

*P*b(*c*)= N

*j*=1

*z*^*jc* (20)

the GREM algorithm and the common EM algorithm after estimating the missing values in the input data set using the unconditional mean imputation method (MENEM) and the nearest neighbour imputation method (NNEM). In the ME- NEM algorithm, the missing values in each feature are re- placed with the mean of the observed values in that feature.

-

l*c* =

-

1

-

N *P*(*c*)

1

N

*E*

*j*=1

XN

*zjc*x*j*|x*o*; R!

!

*j*

(21)

ture that is in the closest feature vector according to the Euclidean distance in the subspace that is composed of the fully observed features. Six data sets are used in this compar-

In the NNEM algorithm, each missing value in a certain fea-

ture vector is replaced with the observed value in the same fea-

*T*

*o*

\_ \_*T*

ison study. The missing values are randomly placed with dif-

R*c* =

- *E*

N *P*(*c*)

*j*=1

*zjc*x*j*x*j* |x*j* ; R

— l*c* l*c* (22)

ferent missing rates in two features of each data set. These features are selected such that the visual separation among

*Step 5:* After convergence, save the resulting FMM and the

total data log-likelihood. Since the EM algorithm is a strict gradient algorithm, it converges slowly to the nearest local maximum of the likelihood function to the starting point [[25].](#_bookmark23) Therefore, the convergence criterion of the EM algo- rithm used in our experiments is identical to the one used by Hunt and Jorgensen [[7]](#_bookmark12), which is to cease iterating when the difference in the log-likelihood function between itera- tions (*t*) and (*t*-10) is less than or equal 10—10. The use of this criterion does not affect the speed of the algorithm so much because the main interest of this algorithm is to handle small data sets.

*Step 6:* Repeat *Step 2–5* several times and then select the best FMM that corresponds to the maximum data log-like- lihood. This repetition is necessary to reduce the sensitivity to the initialisation of the EM algorithm. In our experi- ments, the number of repetitions is chosen arbitrarily to be 20.

*Step 7:* Use the best FMM for 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*–*i*, where *P* (*i*|x) is the probability that x is generated from the component *i*. *Step 8:* Estimate missing values in the data set as explained in *Step 3*.

The LGREM algorithm uses local tuning of the general

data classes is maximum. The mechanism of the occurrence of the missing values in all data sets is missing completely at random (MCAR) [[4]](#_bookmark22). These data sets are described as follows.

* 1. *The first and the second data sets*

These data sets are artificial data sets, each of which contains 150 feature vectors (rows) generated with equal probabilities from three 4D-Gaussian distributions. The mean vectors of these dis-

tributions are l = [ 2 2 2 2 ]*T*; l = [ 4 4 4 4 ]*T*; and

2

1

l = [ 6 6 6 6 ]*T* for the first data set, and l = [ 2 2 2 2 ]*T*;

3 1

l = [ 2 2 6 2 ]*T*; and l = [ 2 2 2 6 ]*T* for the second

3

2

data set. Each one of these Gaussian distributions has a covari- ance matrix R = I4, where I4 is the identity matrix of order four. The difference between both data sets is the correlations between data features. In the first data set, all features are strongly corre- lated, while in the second data set all features are too weakly correlated.

* 1. *The third and the fourth data sets*

These data sets are similar to the first and the second data sets respectively besides an outlier feature vector is added to each one of them. The outlier feature vector in the third data set is

hmax(d ); min(d ); max(d ) ; max(d )i*T*, while the outlier feature vec-

regression (see Eq. [(17)](#_bookmark5)) to produce multiple imputations

1

2

2

3

2

4

tor in the fourth data set is hmax(d ) ; max(d ) i*T*

for each missing value in the data set. Each imputation is ob-

tained via a non-linear multivariate regression using a group

2

1

2

2 ; max(d3); max(d4)

,

of fully observed feature vectors belonging to one of the FMM components. These groups are obtained using Bayes decision rule. This helps the algorithm to be insensitive to global correlations between features of the input data set, and therefore to overcome the limitation of the GREM algo- rithm (see Eq. [(13)](#_bookmark3)) with weakly correlated data. In addition,

where d*i*, *i* = 1:4 are the features of every data set.

* 1. *The fifth data set*

This data set is the Iris data set [[26]](#_bookmark25), which contains 150 feature vectors each of which is a vector in four-feature space. These



Table 1 Comparing different pairs of algorithms using the Student’s paired *t*-test statistic with the first data set.

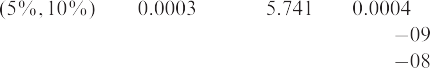


Table 2 Comparing different pairs of algorithms using the Student’s paired *t*-test statistic with the second data set.



Table 3 Comparing different pairs of algorithms using the Student’s paired *t*-test statistic with the third data set.

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 moderate. When each one of these five data sets is used, the missing values are put in the third and in the fourth data features. In addition, each one of the FMMs learned by all algorithms compared in this study consists of three Gaussian components that have non-re- stricted covariance matrices.

* 1. *The sixth data set*

This data set is the Pima Indians Diabetes data set.1 It contains 768 feature vectors each of which is a vector in eight-feature

study consists of two Gaussian components that have non-re- stricted covariance matrices.

The evaluation criterion used in this study to compare the different algorithms is the Mis-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 *Ni*, from which *Nm* feature vectors are not clustered into the component that represents this class in the FMM. Then

*i*

the MCE for class *i* is computed as MCE = *Nm* . Assuming

*i*

*class*

*i Ni*

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 lar-

gely overlapping. Correlations between different pairs of fea-

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* = 1PK MCE*class* .

K

*i*=1

*i*

tures 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 learned by all algorithms compared in this

1 Available at: [<http://www.ics.uci.edu/](http://www.ics.uci.edu/~mlearn/MLSummary.html)~[mlearn/MLSummary.html>](http://www.ics.uci.edu/~mlearn/MLSummary.html)

[Tables 1–5](#_bookmark6) show comparisons of different pairs of the algo- rithms using the Student’s paired *t*-test statistic with each one of the first five 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

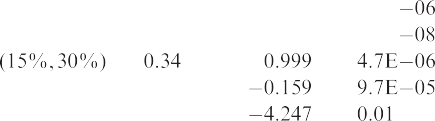


Table 4 Comparing different pairs of algorithms using the Student’s paired *t*-test statistic with the fourth data set.



Table 5 Comparing different pairs of algorithms using the Student’s paired *t*-test statistic with the Iris data set.

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 the third and in the fourth features of each one of the first five data sets. The shaded cells in each table represent the cases in which the difference in performance of certain pairs of algorithms are statistically significant accord- ing to the 5% significance level. [Table 6](#_bookmark7) shows both class and total MCEs when the Pima data set is used with no miss- ing values. [Table 7](#_bookmark9) shows a comparison of different pairs of the

Table 6 The misclassification errors of the FMM learned by the EM algorithm using the Pima data set without missing values.

Data MCE

Class 1 0.426

Class 2 0.511

Total 0.469



Table 7 Comparing different pairs of algorithms using the Student’s paired *t*-test statistic with the Pima data set.



Table 8 Comparing both class-MCEs of each algorithm using the Student’s paired *t*-test statistic with the Pima data set.



Table 9 Comparing different pairs of algorithms using the Student’s paired *t*-test statistic with the Pima data set, while the clustering results are used instead of the true classification of the data set.



Table 10 Comparing both class-MCEs of each algorithm using the Student’s paired *t*-test statistic with the Pima data set, while the clustering results are used instead of the true classification of the data set.

algorithms using the Student’s paired *t*-test statistic with the Pima data set when different pairs of percentages of missing values occur in both of the second and the fifth features. The algorithms are compared using their total MCEs obtained from ten different experiments. [Table 8](#_bookmark10) shows a comparison of both class-MCEs of each algorithm using the Student’s paired *t*-test statistic with the Pima data set. This test examines the statistical significance of the bias of each algorithm in esti- mating the missing values due to the difference in class sizes using pairs of the class-MCEs of the same algorithm obtained from ten different experiments. The shaded cells represent the cases in which the bias of a certain algorithm is statistically sig- nificant according to the 5% significance level. Since both clas- ses of the Pima data set are largely overlapping (see [Table 6](#_bookmark7)), and their sizes are very different, biased estimation of the miss- ing values can affect the learning of the FMM parameters such that low MCE can be obtained. Therefore, it is decided to remove the effect of the overlap between classes on the learn- ing of the FMM parameters. This is achieved by using the clus- tering results, obtained from the FMM trained by the EM algorithm using the Pima data set without missing values, in- stead of the true classification of this data set in computing the MCE. This will clearly show the effect of the estimation of the missing values on the learning of the FMM parameters. The new results of the algorithms compared are shown in [Tables 9 and 10.](#_bookmark11)

1. Discussion of results

[Tables 1–5](#_bookmark6) show that the performance of the LGREM algo- rithm outperforms (*T*-values are negative and *P*-values are less than or equal to 0.05) the performances of the other algo- rithms, especially when there are weak correlations between

data features, few outliers in the data set, or overlap among data classes in the data space. In these cases, local tuning of the general regression used in the LGREM algorithm is less dependent on the overall correlations than the general regres- sion used in the GREM algorithm. Therefore, the LGREM algorithm outperforms the GREM algorithm when features of the data are too weakly correlated (see the results with the second and the fourth data sets in [Tables 2 and 4](#_bookmark6)). In addition, estimating the missing values from the observed feature vectors belonging to each cluster used in the LGREM algorithm is less dependent on the FMM parameters than the maximum likeli- hood estimation used in the MEM algorithm. Therefore, the LGREM algorithm outperforms the MEM algorithm when there are few outliers in the data set, or overlap among data classes in the data space (see the results with the fourth and the Iris data sets in [Tables 4 and 5](#_bookmark8)). This requires each mixture component or cluster in the data space to contain at least one feature vector that is fully observed.

[Table 6](#_bookmark7) shows that the Pima data set has two classes that are largely overlapping. Because of this overlap and the large difference in size between both classes the algorithms that pro- duce biased estimation of the missing values produce less total MCE than the algorithms that produce unbiased estimations. This is shown in [Tables 7 and 8](#_bookmark9). Although [Table 8](#_bookmark10) shows that the LGREM algorithm is unbiased in all pairs of percentages of the missing values [Table 7](#_bookmark9) shows that it produces higher total MCE than other algorithms. Using clustering results of the FMM, learned from the Pima data set with no missing values, instead of the true classification in computing the MCE removes the effect of the class overlap and shows only the effect of the estimation of the missing values on the learn- ing of the FMM parameters. Therefore, [Table 9](#_bookmark11) shows that the LGREM and the NNEM algorithms are statistically similar and they have the smallest total MCEs among all algorithms

in all pairs of percentages of the missing values. [Table 10](#_bookmark11) shows that the LGREM algorithm is superior over the NNEM algorithm because it is unbiased in its estimation of the missing values due to the large difference in class sizes.

In general, local tuning of the general regression causes the LGREM algorithm to outperform the GREM algorithm in the case of too weak global correlations between different pairs of data features. This means that the LGREM algorithm is insen- sitive to global correlations between features of the input data set. In addition, this feature causes the LGREM algorithm to outperform the MEM algorithm in cases such as the occur- rence of some outlier feature vectors or the overlap among data classes in the data space. This conclusion agrees with the results shown in [[27]](#_bookmark26) regarding the preference of the impu- tation techniques that use the pair-wise relations between data features to those techniques that do not. Finally, when the sizes of the data classes are largely different (i.e., unbalanced data) the LGREM algorithm is superior over the other algo- rithms compared with as it produces the most accurate and unbiased estimation of the missing values and the parameters of the FMM, which is used for clustering the input data set. This is because the imputation of the missing values in the LGREM algorithm is both local and independent of FMM parameters.

1. Conclusions

In this paper, the GREM and the MEM algorithms are ana- lysed and their strengths and weaknesses are explained. This analysis leads to the proposal of the LGREM algorithm to overcome problems of both algorithms. A comparison study shows the superiority of the LGREM algorithm over several algorithms commonly used in the literature including the MEM algorithm and the GREM algorithm in unsupervised learning of FMM parameters using data sets with missing val- ues. The LGREM algorithm produces the most accurate and unbiased estimation of the missing values and the best estima- tion of the FMM parameters. This is shown clearly when few outliers occurs in the data set, data classes are overlapped in the data space, or when data classes have large differences in their sizes.

References

1. McLachlan G, Peel D. Finite mixture models. New York: Wiley; 2000.
2. Tra˚ve´n H. A neural network approach to statistical pattern classification by semiparametric estimation of probability density functions. J IEEE Trans Neural Netw 1991;2(3):366–77.
3. Dempster A, Laird N, Rubin D. Maximum likelihood from incomplete data via the EM algorithm (with discussion). J R Stat Soc 1977;B(39):1–38.
4. Little R, Rubin D. Statistical analysis with missing data. New York: John Wiley & Sons; 1987.
5. Ghahramani Z, Jordan M. Supervised learning from incomplete data via an EM approach. In: Cowan J, Tesauro G, Alspector J, editors. Advances in neural information processing systems. San Francisco, CA, USA: Morgan Kaufmann Publishers; 1994. p. 120–7.
6. Hunt L. Clustering using finite mixture models. Ph.D. thesis. New Zealand: Department of Statistics, University of Waikato; 1996.
7. Hunt L, Jorgensen M. Mixture model clustering for mixed data with missing information. J Comput Stat Data Anal 2003;41:429–40.
8. Dybowski R. Classification of incomplete feature vectors by radial basis function networks. J Pattern Recognit Lett 1998;19: 1257–64.
9. Morris A, Cooke M. Some solutions to the missing feature problem in data classification, with application to noise robust ASR. In: Proceeding of the international conference on acoustics speech and signal processing, Seattle, WA, USA; 1998. p. 737–40.
10. Domma F, Ingrassia S. Mixture models for maximum likelihood estimation from incomplete values. In: Borra S, Rocci R, Vichi M, Schader M, editors. Studies in classification, data analysis and knowledge organization. Berlin: Springer-Verlag; 2001. p. 201–8.
11. Yoon S, Lee S. Training algorithm with incomplete data for feed- forward neural networks. J Neural Process Lett 1999;10:171–9.
12. Troyanskaya O, Cantor M, Sherlock G, Brown P, Hastie T, Tibshirani R, Botstein D, Altman R. Missing value estimation methods for DNA microarrays. J Bioinform 2001;17(6):520–5.
13. Lin TI, Lee JC, Ho HJ. On fast supervised learning for normal mixture models with missing information. J Pattern Recognit 2006;39:1177–87.
14. Bouveyron C, Girard S. Robust supervised classification with mixture models: learning from data with uncertain labels. J Pattern Recognit 2009;42:2649–58.
15. Specht D. A general regression neural network. J IEEE Trans Neural Netw 1991;2(6):568–76.
16. Tresp V, Ahmad S, Neuneier R. Training neural networks with deficient data. In: Cowan J, Tesauro G, Alspector J, editors. Advances in neural information processing systems, vol. 6. San Mateo, CA: Morgan Kaufman; 1994. p. 128–35.
17. Tresp V, Neuneier R, Ahmad S. Efficient methods for dealing with missing data in supervised learning. In: Tesauro G, Tour- etzky D, Leen T, editors. Advances in neural information processing systems, vol. 7. Cambridge, MA: MIT Press; 1995. p. 689–96.
18. Raymond M. Missing data in evaluation research. J Eval Health Prof 1986;9:395–420.
19. Raymond M, Roberts D. A comparison of methods for treating incomplete data in selection research. J Educ Psychol Meas 1987;47:13–26.
20. Zhou X, Wang X, Dougherty E. Missing-value estimation using linear and non-linear regression with Bayesian gene selection. J Bioinform 2003;19(17):2302–7.
21. Guo P, Chen C, Lyu M. Cluster number selection for a small set of samples using the Bayesian Ying–Yang model. J IEEE Trans Neural Netw 2002;13(3):757–63.
22. Milligan GW, Cooper MC. A study of standardization of variables in cluster analysis. J Classif 1988;5:181–204.
23. Schaffer CM, Green PE. An empirical comparison of variable standardization methods in cluster analysis. J Multivariate Behav Res 1996;31(2):149–67.
24. Vlassis N, Likas A. A greedy EM algorithm for Gaussian mixture learning. J Neural Process Lett 2002;15:77–87.
25. Yin H, Allinson NM. Comparison of a Bayesian SOM with the EM algorithm for Gaussian mixtures. In: Proceeding of workshop on self-organising maps (WSOM’97); 1997. p. 118–23.
26. Fisher R. The use of multiple measurements in taxonomic problems. Annu Eugenics 1936;7:179–88.
27. Huisman M. Imputation of missing item responses: some simple techniques. J Qual Quant 2000;34:331–51.