PERGAMON

Pattern Recognition 35 (2002) 1127–1142

PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY

www.elsevier.com/locate/patcog

Robust clustering by deterministic agglomeration EM of mixtures of multivariate *t*-distributions

Shy Shoham*

Department of Bioengineering, University of Utah, 20 S. 2030 E., Rm. 506, Salt Lake City, Utah 84112-9458, USA

Received 16 August 2000; accepted 5 March 2001

Abstract

This paper presents new robust clustering algorithms, which significantly improve upon the noise and initialization sensitivity of traditional mixture decomposition algorithms, and simplify the determination of the optimal number of clusters in the data set. The algorithms implement maximum likelihood mixture decomposition of multivariate *t*-distributions, a robust parametric extension of gaussian mixture decomposition. We achieve improved convergence capability relative to the expectation–maximization (EM) approach by deriving deterministic annealing EM (DAEM) algorithms for this mixture model and turning them into agglomerative algorithms (going through a monotonically decreasing number of components), an approach we term deterministic agglomeration EM (DAGEM). Two versions are derived, based on two variants of DAEM for mixture models. Simulation studies demonstrate the algorithms' performance for mixtures with isotropic and non-isotropic covariances in two and 10 dimensions with known or unknown levels of outlier contamination. © 2002 Pattern Recognition Society. Published by Elsevier Science Ltd. All rights reserved.

Keywords: Clustering; Finite mixture models; EM algorithm; Robust algorithms; *t*-distribution; Deterministic annealing; Agglomerative algorithms.

1. Introduction

The use of finite mixture model fitting as a systematic and flexible approach to clustering of multivariate data has become common practice in statistics [1], as well as in a large number of pattern recognition and analysis applications [2]. Many authors use the expectation–maximization (EM) algorithm [3,4] for the task of maximizing the model likelihood, with most work in the field focusing on mixtures of multivariate gaussians. However, it is widely acknowledged that this framework suffers from three significant difficulties, which are shared by most other traditional algorithms for the fuzzy partition of data. First, EM is a *local* maximum seeker, which, in view of the mixture likelihood being riddled with many local maxima, makes its result highly sensitive to initialization. Second, the problem of determining the optimal number of components in the mixture remains difficult, typically requiring multiple runs of the algorithm with different numbers of components. The third problem is the lack of robustness with respect to outliers that results from using gaussians, often, a poor model of variability in real data sets.

In this paper we introduce a new method for robust model-based clustering, with convergence properties that significantly surpass those of the ordinary EM algorithm. Our strategy combines two recent approaches to achieving improved convergence properties within the EM framework: deterministic annealing [5] and competitive agglomeration [6], by running a deterministic annealing EM algorithm in agglomeration mode. Robustness is enhanced by moving from mixture components that are multivariate gaussians, to multivariate *t*-distributions with their wider tails, a standard approach

* Tel.: +1-801-581-3817; fax: +1-801-581-8966.
E-mail address: shay@eng.utah.edu (S. Shoham).

PII: S0031-3203(01)00080-2

Nomenclature

β	inverse temperature
g	number of components in the mixture
N	number of data points
p	dimensionality of data
π_j, $\boldsymbol{\mu}_j$, $\boldsymbol{\Sigma}_j$	mixing proportion, mean and covariance of component j
z_{ij}	membership of data point i with respect to component j ($0 \leqslant z_{ij} \leqslant 1$)
u_{ij}	weights indicating 'typicality' of data point i w.r.t. component j ($u_{jj} \ll 1$ for outliers)
k	iteration #
$\boldsymbol{x}_{obs}$	observed data
$\boldsymbol{x}_{mis}$	'missing' data. Includes memberships and weights of data points
$f(\boldsymbol{x}_{mis})$	probability density of missing data
F_β, F'_β	free energy functions
ψ	Digamma function
$\delta(\boldsymbol{x},\ \boldsymbol{\mu}_j;\ \boldsymbol{\Sigma})$	mahalanobis distance between $\boldsymbol{x}_i$ and $\boldsymbol{\mu}_j$ squared
v	degrees of freedom parameter (DOF)
Bold	matrix/vector. Plain-scalar quantity

with probabilistic models [7,8]. As a direct result of using the t-distributions an additional set of weights that automatically down-weighs atypical data points is introduced, thus achieving the desired robustness.

The rest of the paper is organized as follows: first, we review related work in the field. In Section 2 we present the mixture of multivariate t-distributions model, repeat the derivation of the EM algorithm for this model, and develop 2 versions of the DAEM algorithm for this model. In Section 3 we present significant problems encountered by these algorithms when applied as prescribed by previous studies of DAEM. We present a simple yet powerful modification of the annealing schedule, termed DAGEM, which overcomes these problems. In Section 4 we test the algorithms on two more benchmark problems. We conclude in Section 5 with a general discussion of several aspects of the new algorithms, as well as related issues left open.

1.1. Related work

A number of recent studies have attempted to resolve one or more of the problems with the gaussian mixture EM framework listed above.

1.1.1. Initialization sensitivity

Deterministic annealing, split and merge operations and competitive agglomeration are methods that explicitly attempt to overcome the initialization sensitivity problem. All three methods essentially add an extra schedule on top of the expectation and maximization iterations of the EM algorithm.

Deterministic annealing EM (DAEM) was introduced by Ueda and Nakano [5], and a modified version (termed REM 2) was independently developed by Sahani [9]. Applications of deterministic annealing are reviewed in [10]. The standard EM algorithm [3,4] can be viewed [11] as alternating maximization of

$$F = \int \log(p(\boldsymbol{x}_{obs} \mid \boldsymbol{x}_{mis}; \theta)\, p(\boldsymbol{x}_{mis}; \theta))\, f(\boldsymbol{x}_{mis})\, \mathrm{d}\boldsymbol{x}_{mis} - \int (\log f(\boldsymbol{x}_{mis}))\, f(\boldsymbol{x}_{mis})\, \mathrm{d}\boldsymbol{x}_{mis},$$

first with respect to the distribution of the missing data, $f(\boldsymbol{x}_{mis})$ (E step), and then with respect to the model's parameters, θ (M step). Similarly, DAEM is based on alternating maximization of a more general 'negative free energy'

$$F_\beta = \beta \int \log(p(\boldsymbol{x}_{obs} \mid \boldsymbol{x}_{mis}; \theta)\, p(\boldsymbol{x}_{mis}; \theta))\, f(\boldsymbol{x}_{mis})\, \mathrm{d}\boldsymbol{x}_{mis} - \int (\log f(\boldsymbol{x}_{mis}))\, f(\boldsymbol{x}_{mis})\, \mathrm{d}\boldsymbol{x}_{mis},$$

where the parameter β has the intuitive interpretation of 1/temperature, and for $\beta = 1$ the DAEM iterations are equivalent to those of the EM. β undergoes an annealing schedule, changing from an extremely small value ('infinite temperature') to 1, while at each intermediate value the modified E and M steps are repeated to convergence. At lower values of β the negative free energy is increasingly convex (totally convex entropy term at $\beta = 0$), and thus less riddled with local maxima. The motivation is therefore an attempt to track the global

maximum, in progressively less convex models, an idea related to Homotopy Continuation methods of optimization [12]. When applied to mixture models, the algorithm undergoes a series of phase transitions with decreasing 'temperature', in which the model size is increased (starting with one component at low β). Sahani [9] notes that the DAEM algorithm undergoes problematic jumps in the free energy when a mixture component is split. To correct this problem, he introduces a slightly different 'negative free energy' term

$$F_\beta = \int (\beta \log(p(\boldsymbol{x}_{obs} \mid \boldsymbol{x}_{mis}; \theta)) + \log(p(\boldsymbol{x}_{mis}; \theta)))f(\boldsymbol{x}_{mis})\,\mathrm{d}\boldsymbol{x}_{mis} - \int \log(f(\boldsymbol{x}_{mis}))f(\boldsymbol{x}_{mis})\,\mathrm{d}\boldsymbol{x}_{mis}.$$

In general, the DAEM framework is incapable of tracking the global maximum, and in particular in mixture models the global maximum does not move smoothly as the temperature varies, a condition for guaranteed tracking. A series of simulation experiments in Refs. [5,9] have shown, however, that DAEM and REM 2 for mixtures of gaussians have a significantly better convergence property than those of the standard EM.

Ueda et al. [13] note, that EM and DAEM for mixture have a difficulty avoiding local maxima associated with situations where separated areas of space have too few or too many components. To overcome this apparent problem they suggest allowing some mixture components to merge while others split (total number of components is conserved), a process they call split and merge EM (SMEM). Numerical results of the convergence capabilities of this algorithm are very encouraging. Competitive agglomeration starts with a highly over-specified number of components, and adds a penalty term to the log-likelihood that increasingly favors agglomeration of all the data points into fewer, larger components. Marginally small components are discarded, and following the agglomeration phase, the weight of the penalty term is progressively reduced. In different publications by Frigui et al. [6,14,15] the penalty terms $\alpha \sum_{j=1}^{g} \pi_j^2$ and $\alpha \sum_{j=1}^{g} \pi_j \log \pi_j$ are used, with α a parameter that undergoes an annealing schedule (increase from zero to a finite value, followed by a gradual decrease back to zero). A related Bayesian approach with the penalty term $N_C/2 \sum_{j=1}^{g} \log \pi_j$ (where N_C represents the number of parameters per mixture component) was introduced in Ref. [16]. In competitive agglomeration the update of the mixing proportions π_j during the M step is modified, while in DAEM type algorithms the E step is the one modified.

In addition to these methods, other authors have introduced adaptations of methods of global optimization to the EM framework, e.g. genetic optimization [17].

1.1.2. Optimal model size

Determination of the optimal number of components in a mixture model traditionally depends on finding the ML models for all relevant number of components, and then picking the one that has maximal *penalized* log-likelihood, according to a selected penalty criterion, such as the AIC, BIC, MDL or other criteria [18,19].

Several authors have attempted to make the process of moving between different model sizes more efficient. Competitive agglomeration automatically goes through a monotonically decreasing number of mixture components, however, it does not include a natural stopping criteria related to penalized log-likelihood, relying instead on predetermination of an exact annealing schedule. In contrast, the agglomerative EM based algorithm of Banfield and Raftery [20] and Fraley [21], which also goes through a monotonically decreasing number of components (selecting at each model size the optimal components to be merged), allows for a penalized log-likelihood comparison between different model sizes. An Agglomerative EM algorithm with a Bayesian penalty term was proposed in Ref. [19], and other Bayesian approaches based on trimming unimportant components were described in Refs. [16,22].

DAEM and REM 2 go through a monotonically *increasing* number of components, changing during phase transitions. To determine which size is optimal, one would have to anneal models of all sizes, each to $\beta = 1$. With REM 2, Sahani [9] uses a monotonicity argument to efficiently purge many of the possibilities while annealing, a process he terms 'Cascading Model Selection'.

1.1.3. Robustness

One can roughly separate the attempts at enhancing the robustness of the EM gaussian mixture decomposition algorithm into parametric and other methods. A review of many robust clustering methods appears in Ref. [23].

Parametric methods involve choosing a parametric model to replace one based on a mixture of gaussians, generally one where one or more of the mixture's components can 'explain' outliers. Perhaps the most popular approach is adding an extra component with uniform density [9,20], diffusely covering the entire measurement space. The mixing proportion of the uniform component becomes an additional mixture parameter. A second approach is to choose mixture components with wider tails than the gaussian distribution. A standard choice in statistics is the multivariate t distribution [7,24]. An EM algorithm for mixtures of t-distributions was introduced recently in Ref. [8]. This algorithm introduces an additional set of weights, estimated during the E step, which essentially performs a soft rejection of outliers. A significant advantage of using the t distribution is the ability to tune the model's robustness to a particular

application or even a particular data set, by tuning the degrees of freedom parameter.

Semi-parametric approaches, such as the use of various M-estimators, borrow from Huber's work on robust statistics [25]. In particular, the use of Huber's ψ-function [26,27] is a hybrid of a gaussian distribution with laplacian tails. Another approach introduced recently uses least trimmed squares estimators [15].

2. Deterministic annealing EM for multivariate *t* mixtures

2.1. The multivariate t mixture model

The components in our mixture are multivariate t-distributions, which are parameterized by a unique mean $\boldsymbol{\mu}_j$, covariance matrix $\boldsymbol{\Sigma}_j$ and a 'degrees of freedom' (DOF) parameter v (same for all components). Effectively, v parameterizes the 'robustness' of the distribution, that is, how wide the tails are. The case $v \to \infty$ corresponds to a gaussian distribution and when $v = 1$ we obtain the wide tailed multivariate Cauchy distribution (the covariance is infinite for $v \leqslant 2$). For p-dimensional data vectors $\boldsymbol{x}_i$, the multivariate t distribution's p.d.f. is

$$p(\boldsymbol{x}_i \mid \theta_j) = p(\boldsymbol{x}_i, \mid \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j, v) = \frac{\Gamma((v+p)/2)|\boldsymbol{\Sigma}_j|^{-1/2}}{\Gamma(\frac{1}{2})\Gamma(v/2)v^{p/2}} \times \frac{1}{[1+(\delta(\boldsymbol{x}_i, \boldsymbol{\mu}_j; \boldsymbol{\Sigma}_j)/v)]^{(v+p)/2}}, \tag{1}$$

where $\delta(\boldsymbol{x}_i, \boldsymbol{\mu}_j; \boldsymbol{\Sigma}_j) = (\boldsymbol{x}_i - \boldsymbol{\mu}_j)^{\mathrm{T}} \boldsymbol{\Sigma}_j^{-1} (\boldsymbol{x}_i - \boldsymbol{\mu}_j)$ is the Mahalanobis distance between $\boldsymbol{x}_i$ and $\boldsymbol{\mu}_j$ squared.

A two-step method of generating these t-distributed vectors, is by generating samples from a multivariate gaussian distribution with mean $\boldsymbol{\mu}_j$ and covariance matrix $\boldsymbol{\Sigma}_j/u$:

$$p(\boldsymbol{x}_i \mid u_i) \sim N(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j/u_i) = \frac{1}{(2\pi/u_i)^{p/2}|\boldsymbol{\Sigma}_j|^{1/2}} \times \exp\left(-\frac{u_i}{2}\delta(\boldsymbol{x}_j, \boldsymbol{\mu}_j; \boldsymbol{\Sigma}_j)\right), \tag{2}$$

where the random scalar u_i is generated from a gamma distribution $\gamma(u_i; v/2, 2/v)$ distribution, given by

$$\gamma(u_i; \alpha, \beta) = \frac{1}{\beta^{\alpha}\Gamma(\alpha)} \times u_i^{\alpha-1} \exp(-u_i/\beta), \; u_i, \alpha, \beta > 0. \tag{3}$$

We assume that the observed N data points are drawn randomly, independently and are identically distributed according to a mixture of multivariate t-distributions, where mixture component j has the mixing proportion π_j. The data log-likelihood is

$$L = \log\left(\prod_{i=1}^{N}\left(\sum_{j=1}^{g}\pi_j\, p(\boldsymbol{x}_i \mid \theta_j)\right)\right) = \sum_{i=1}^{N}\log\left(\sum_{j=1}^{g}\pi_j\, p(\boldsymbol{x}_i \mid \theta_j)\right). \tag{4}$$

We are now interested in solving the maximum likelihood problem, that is, in finding the parameters $\theta \equiv \{\pi_{1\ldots g}, \boldsymbol{\mu}_{1\ldots g}, \boldsymbol{\Sigma}_{1\ldots g}, v\}$ that maximize this log-likelihood under the constraint $\sum_{j=1}^{g}\pi_j = 1$, a problem which does not have a closed form solution.

2.2. EM for mixtures of multivariate t-distributions

In this section (and in Appendix A) we repeat for the sake of completeness and for the reader's convenience the derivation of the EM algorithm for mixtures of t-distributions, the algorithm was first presented in Ref. [8].

EM consists of the following two steps:

E step: Evaluate the Q function: $Q(\theta) = E_{f(\boldsymbol{x}_{mis})}[\log p(\boldsymbol{x}_{obs} \mid \boldsymbol{x}_{mis}; \theta)]$ with:

$$f(\boldsymbol{x}_{mis}) = p(\boldsymbol{x}_{mis} \mid \boldsymbol{x}_{obs}; \theta^{(t)}) = \frac{p(\boldsymbol{x}_{mis}; \theta^{(t)})\, p(\boldsymbol{x}_{obs} \mid \boldsymbol{x}_{mis}; \theta^{(t)})}{\int p(\boldsymbol{x}_{mis}; \theta^{(t)})\, p(\boldsymbol{x}_{obs} \mid \boldsymbol{x}_{mis}; \theta^{(t)})\, \mathrm{d}\boldsymbol{x}_{mis}}, \tag{5}$$

M step: maximize $Q(\theta)$ with respect to θ.

We proceed by augmenting the observed data with 'missing data' to obtain the complete data: $\{\boldsymbol{x}_i, z_{ij}, u_i;\ i = 1..N,\ j = 1..g\}$, where $\boldsymbol{x}_i$ is the ith observed data vector and the missing data consists of: z_{ij}, a binary indicator variable which takes the value 1 when $\boldsymbol{x}_i$ came from mixture component j (zero otherwise), and u_i, a gamma distributed random scalar. The respective probabilities are:

$$p(\boldsymbol{x}_i \mid z_{ij} = 1, u_i, \boldsymbol{\mu}_j, \boldsymbol{\Sigma}, v) \sim N(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}/u_i) = \frac{1}{(2\pi/u_i)^{p/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\frac{u_i}{2}\cdot\delta(\boldsymbol{x}_i, \boldsymbol{\mu}_j; \boldsymbol{\Sigma})\right), \tag{6}$$

$$p(u_i \mid v) = \gamma\left(\frac{v}{2}, \frac{2}{v}\right) = \frac{1}{(2/v)^{v/2}\Gamma(v/2)} \times u_i^{v/2-1}\exp\left(-\frac{v}{2}u_i\right), \quad u_i > 0, \tag{7}$$

$$p(z_{ij} = 1) = \pi_j. \tag{8}$$

We substitute these probabilities in $Q(\theta)$:

$$
\begin{aligned}
&Q(\theta)\\
&= E_{z,u}\left[\sum_{i=1}^{N}\sum_{j=1}^{g} z_{ij}[\log(\, p(\boldsymbol{x}_i \mid u_i, z_{ij}, \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j, v)\, p(z_{ij})\right.\\
&\qquad \left.\times p(u_i \mid v))]\right]\\
&= E_{z,u}\left[\sum_{i=1}^{N}\sum_{j=1}^{g} z_{ij}\left[-\frac{u_i}{2}\delta(\boldsymbol{x}_i, \boldsymbol{\mu}_j; \boldsymbol{\Sigma}_j) - \frac{1}{2}\log|\boldsymbol{\Sigma}_j| + \log \pi_j\right.\right.\\
&\qquad \left.\left.+\frac{v}{2}\left(\log u_i - u_i + \log\frac{v}{2}\right) - \log \Gamma(v/2) + C\right]\right], \qquad (9)
\end{aligned}
$$

where the expectation is with respect to the missing data probability $f(z,u)$, and C is a constant, independent of the parameters (and will therefore not affect the M step). Intuitively, multiplication by z_{ij} assures that only the component j to which the point $\boldsymbol{x}_i$ belongs ($z_{ij}=1$) contributes to $Q(\theta)$. This expression is clearly linear in z_{ij}. To achieve linearity in u_i as well, which will significantly simplify the algorithm, we turn to a new parameter set: $\theta' \equiv \{\pi_{1\ldots g}, \boldsymbol{\mu}_{1\ldots g}, \boldsymbol{\Sigma}_{1\ldots g}\}$, essentially assuming that v is a known constant. Now we have

$$
\begin{aligned}
Q(\theta') = E_{z,u}&\left[\sum_{i=1}^{N}\sum_{j=1}^{g} z_{ij}\left[-\left(\frac{u_i}{2}(\boldsymbol{x}_i - \boldsymbol{\mu}_j)^{\mathrm{T}}\boldsymbol{\Sigma}_j^{-1}(\boldsymbol{x}_i, \boldsymbol{\mu}_j)\right)\right.\right.\\
&\left.\left.-\frac{1}{2}\log|\boldsymbol{\Sigma}_j| + \log \pi_j + C'\right]\right]. \qquad (10)
\end{aligned}
$$

The EM steps for the problem defined by Eqs. (6)–(8), (10) are derived in Appendix A. The resulting learning steps at iteration k are:

E step:

$$
\hat{z}_{ij} = E_{f(\boldsymbol{x}_{mis})}(z_{ij}) = \frac{\pi_j\, p(\boldsymbol{x}_i \mid \boldsymbol{\mu}_j^{(k-1)}, \boldsymbol{\Sigma}_j^{(k-1)}, v)}{\sum_{j=1}^{g} \pi_j\, p(\boldsymbol{x}_i \mid \boldsymbol{\mu}_j^{(k-1)}, \boldsymbol{\Sigma}_j^{(k-1)}, v)} \qquad (11)
$$

with $p(\boldsymbol{x}_i \mid \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j, v)$ defined in Eq. (1).

$$
\hat{u}_{ij} \equiv \frac{p+v}{\delta(\boldsymbol{x}_i, \boldsymbol{\mu}_j^{(k-1)}; \boldsymbol{\Sigma}_j^{(k-1)}) + v}. \qquad (12)
$$

M step:

$$
\pi_j = \frac{\sum_{i=1}^{N} \hat{z}_{ij}}{N} \qquad (13)
$$

$$
\boldsymbol{\mu}_j^{(k)} = \frac{\sum_{i=1}^{N} \hat{z}_{ij}\hat{u}_{ij}\boldsymbol{x}_i}{\sum_{i=1}^{N} \hat{z}_{ij}\hat{u}_{ij}}, \qquad (14)
$$

$$
\boldsymbol{\Sigma}_j^{(k)} = \frac{\sum_{i=1}^{N} (\hat{z}_{ij}\hat{u}_{ij})(\boldsymbol{x}_i - \boldsymbol{\mu}_j^{(k)})(\boldsymbol{x}_i - \boldsymbol{\mu}_j^{(k)})^{\mathrm{T}}}{\sum_{i=1}^{N} \hat{z}_{ij}}. \qquad (15)
$$

It was shown in Ref. [28] that changing the denominator in the covariance update yields significantly faster convergence for an EM algorithm for the estimation of the parameters of a *single* multivariate t-distribution. The different update was motivated in several ways Refs. [28–31]. For the mixture model, the equivalent update is

$$
\boldsymbol{\Sigma}_j^{(k)} = \frac{\sum_{i=1}^{N} (\hat{z}_{ij}\hat{u}_{ij})(\boldsymbol{x}_i - \boldsymbol{\mu}_j^{(k)})(\boldsymbol{x}_i - \boldsymbol{\mu}_j^{(k)})^{\mathrm{T}}}{\sum_{i=1}^{N} \hat{z}_{ij}\hat{u}_{ij}}. \qquad (16)
$$

Or, in the case we constrain all components to have a common covariance:

$$
\boldsymbol{\Sigma}^{(k)} = \frac{\sum_{i=1}^{N}\sum_{j=1}^{g} (\hat{z}_{ij}\hat{u}_{ij})(\boldsymbol{x}_i - \boldsymbol{\mu}_j^{(k)})(\boldsymbol{x}_i - \boldsymbol{\mu}_j^{(k)})^{\mathrm{T}}}{\sum_{i=1}^{N}\sum_{j=1}^{g} \hat{z}_{ij}\hat{u}_{ij}}. \qquad (17)
$$

2.3. REM 2 for mixtures of multivariate t-distributions

The REM 2 algorithm (mentioned in the introduction) has the following modified negative free energy [9]

$$
\begin{aligned}
F'_{\beta}(\theta) = E_{f(\boldsymbol{X}_{mis})}&[\beta \log\, p(\boldsymbol{x}_{obs} \mid \boldsymbol{x}_{mis}; \theta) + \log\, p(\boldsymbol{x}_{mis}; \theta)]\\
&-E_{f(\boldsymbol{X}_{mis})}[\log(f(\boldsymbol{x}_{mis}))]. \qquad (18)
\end{aligned}
$$

The first term, a 'temperature dependent' version of the Q function will be termed Q'_{β}. Substituting the probabilities (6)–(8) yields the 'temperature dependent' equivalent of Eq. (10):

$$
\begin{aligned}
&Q'_{\beta}(\theta')\\
&= E_{z,u}\left[\sum_{i=1}^{N}\sum_{j=1}^{g} z_{ij}\left[-\left(\frac{\beta u_i}{2}(\boldsymbol{x}_i - \boldsymbol{\mu}_j)^{\mathrm{T}}\boldsymbol{\Sigma}_j^{-1}(\boldsymbol{x}_i - \boldsymbol{\mu}_j)\right)\right.\right.\\
&\qquad \left.\left.-\frac{\beta}{2}\log|\boldsymbol{\Sigma}_j| + \log \pi_j + C'\right]\right]. \qquad (19)
\end{aligned}
$$

The E step is derived in Appendix B (the M step is identical to the regular one). The resulting updates are:

$$
\begin{aligned}
\hat{z}_{ij} &= E_{f(\boldsymbol{X}_{mis})}(z_{ij})\\
&= \frac{\pi_j|\boldsymbol{\Sigma}_j|^{-\beta/2}(\beta\delta(\boldsymbol{x}_i, \boldsymbol{\mu}_j^{(k-1)}; \boldsymbol{\Sigma}_j^{(k-1)}) + v)^{-(\beta p+v)/2}}{\sum_{j=1}^{g} \pi_j|\boldsymbol{\Sigma}_j|^{-\beta/2}(\beta\delta(\boldsymbol{x}_i, \boldsymbol{\mu}_j^{(k-1)}; \boldsymbol{\Sigma}_j^{(k-1)}) + v)^{-(\beta p+v)/2}}, \qquad (20)
\end{aligned}
$$

$$
\hat{u}_{ij} \equiv \frac{\beta p + v}{\beta\delta(\boldsymbol{x}_i, \boldsymbol{\mu}_j^{(k-1)}; \boldsymbol{\Sigma}_j^{(k-1)}) + v}. \qquad (21)
$$

We observe that the term $(\beta\delta(\boldsymbol{x}_i,\boldsymbol{\mu}_j;\boldsymbol{\Sigma}_j)+v)^{-(\beta p+v)/2}$ corresponds effectively to a t-distribution with covariance $\boldsymbol{\Sigma}_j/\beta$, and $v-(1-\beta)p$ degrees of freedom, both terms contributing to a 'stretching' of the distribution and its tails.

2.4. Modified DAEM for mixtures of multivariate t-distributions

As noted in the introduction, the problem with Ueda and Nakano's DAEM algorithm is the fact that the negative free energy is not constant when a single component is split into two identical components. Below we suggest a new, conceptually simpler, resolution of this problem that relies on modifying the constraints on the mixing proportions, rather than the 'negative free energy'.

The negative DAEM free energy [5] is slightly different from that of REM2:

$$\begin{aligned}F_\beta &\equiv E_{f(\boldsymbol{X}_{mis})}[\beta\log(p(\boldsymbol{x}_{obs}\mid\boldsymbol{x}_{mis};\theta)p(\boldsymbol{x}_{mis};\theta))]\\ &\quad -E_{f(\boldsymbol{X}_{mis})}[\log(f(\boldsymbol{x}_{mis}))]\\ &= \log\int((p(\boldsymbol{x}_{obs}\mid\boldsymbol{x}_{mis};\theta)p(\boldsymbol{x}_{mis};\theta))^\beta)\,\mathrm{d}\boldsymbol{x}_{mis}\\ &= \sum_{i=1}^{N}\log\sum_{j=1}^{g}\pi_j^\beta(p(\boldsymbol{x}\mid z_{ij},u_i;\theta'^{(t)})p(u_i\mid v))^\beta. \qquad (22)\end{aligned}$$

According to Eq. (22), when a component is split to two components with identical mean and covariance, the new components' mixing proportion has to obey: $\pi^\beta=\pi_1^\beta+\pi_2^\beta$ in order to assure F_β constancy. However, this update will violate the constraint: $\sum_{j=1}^{g}\pi_j=1$ which is typically imposed. An alternative constraint, which will not be violated by such a replacement, is

$$\sum_{j=1}^{g}\pi_j^\beta=1. \qquad (23)$$

Although this new constraint may seem somewhat awkward at first glance, imposing it allows us to preserve the monotonic likelihood climbing nature of the EM algorithm, as well as the exact correspondence with the original EM problem for $\beta=1$. In fact, introducing a 'new mixing proportion': $\pi'_j=\pi_j^\beta$ yields a new condition for component split: $\pi'=\pi'_1+\pi'_2$ as well as the familiar constraint

$$\sum_{j=1}^{g}\pi'_j=1. \qquad (24)$$

In Appendix B, we derive the E and M steps of this DAEM algorithm, and find that the M step is unchanged, including

$$\pi'_j=\frac{\sum_{i=1}^{N}\hat{z}_{ij}}{N}. \qquad (25)$$

This constraint-modified DAEM approach would yield identical results to Sahani's modified free-energy REM 2 method, in the mixture of gaussians case. However, in the mixture of t-distributions considered in this paper, with 2 sets of augmented data (z and u), the results are different, with the following E step:

$$\hat{z}_{ij}=\frac{\pi'_j|\boldsymbol{\Sigma}_j|^{-\beta/2}(\delta(\boldsymbol{x}_i,\boldsymbol{\mu}_j;\boldsymbol{\Sigma}_j)+v)^{-\beta(p+v-2+2/\beta)/2}}{\sum_{j=1}^{g}\pi'_j|\boldsymbol{\Sigma}_j|^{-\beta/2}(\delta(\boldsymbol{x}_i,\boldsymbol{\mu}_j;\boldsymbol{\Sigma}_j)+v)^{-\beta(p+v-2+2/\beta)/2}}, \qquad (26)$$

$$\hat{u}_{ij}\equiv\frac{p+v-2+2/\beta}{\delta(\boldsymbol{x}_i,\boldsymbol{\mu}_j;\boldsymbol{\Sigma})+v}. \qquad (27)$$

2.5. Degrees of freedom

As with other algorithms based on multivariate t-distributions, we anticipate two types of applications of our method: one where the degrees of freedom are assumed known a priori, and one where it will be estimated together with the rest of the parameters. For the latter case, we compute below an additional M step (maximizing Q_β with respect to v) for our modified DAEM algorithm. We rewrite Q_β with the v-dependent elements included:

$$Q_\beta(\theta)=E_{z,u}\left[\beta\sum_{i=1}^{N}\sum_{j=1}^{g}z_{ij}\left[-\frac{u_i}{2}\delta(\boldsymbol{x}_i,\boldsymbol{\mu}_j;\boldsymbol{\Sigma}_j)-\frac{1}{2}\log|\boldsymbol{\Sigma}_j|+\log\pi_j+\frac{v}{2}\left(\log u_i-u_i+\log\frac{v}{2}\right)-\log\Gamma(v/2)\right]\right]. \qquad (28)$$

This term is no longer linear with respect to u_i. We use the following result:

$$E_{z,u}[z_{ij}\log u_i]=\hat{z}_{ij}\cdot\left(\psi\left(\frac{\beta(p+v-2+2/\beta)}{2}\right)+\log\left(\frac{2}{\beta(\delta(\boldsymbol{x}_i,\boldsymbol{\mu}_j;\boldsymbol{\Sigma})+v)}\right)\right)\equiv\hat{z}_{ij}\cdot\hat{l}_{ij}, \qquad (29)$$

which is easily derived from Eqs. [C.3], [20] and the following integral identity:

$$\int_0^\infty\log u_i\cdot u_i^{a-1}\exp(-u_i/b)\,\mathrm{d}u_i=\Gamma(a)\cdot b^a(\psi(a)+\log(b)),$$

where the digamma function, $\psi(a)$ is defined by

$$\psi(a) \equiv \frac{d(\log \Gamma(a))}{da} = \frac{\Gamma(a)}{\Gamma'(a)}.$$

Finally, we obtain the appropriate M step

$$\frac{d}{dv}[Q_\beta] = \frac{\beta}{2}\sum_{i=1}^{N}\sum_{j=1}^{g}\hat{z}_{ij}\left[\hat{l}_{ij} - \hat{u}_{ij} + \log\frac{v}{2} + 1 - \psi\left(\frac{v}{2}\right)\right] = 0. \quad (30)$$

As Eq. (30) is nonlinear, most authors who considered the unknown DOF problem (e.g. Refs. [7,24,28]) solve it by a search over v. In an attempt to reduce computational load we found the following direct approximation to be extremely accurate ($|v - v^*| < 10^{-3}$):

$$v^* = \frac{2}{y + \log y - 1} + 0.0416\left(1 + \operatorname{erf}\left(0.6594 \log\left(\frac{2.1971}{y + \log y - 1}\right)\right)\right),$$

$$y \equiv -\sum_{i=1}^{N}\sum_{j=1}^{g}\hat{z}_{ij}[\hat{l}_{ij} - \hat{u}_{ij}]/N. \quad (31)$$

A similar derivation using the REM 2 energy function yields the only difference:

$$\hat{l}_{ij} = \psi\left(\frac{\beta p + v}{2}\right) + \log\left(\frac{2}{\beta\delta(\boldsymbol{x}_i, \boldsymbol{\mu}_j; \boldsymbol{\Sigma}) + v}\right). \quad (32)$$

3. Annealing schedule

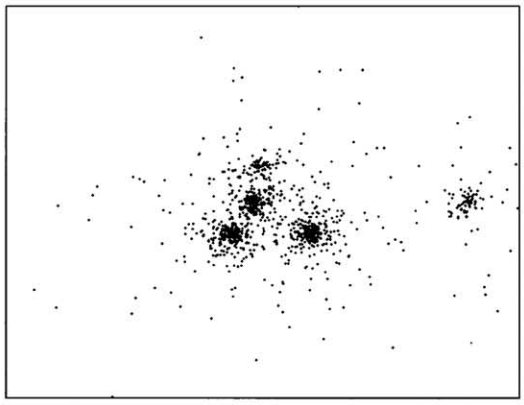
3.1. Classical approach

In previous publications involving DAEM, the annealing parameter β was varied from an extremely small value (high 'temperature'), where the mixture collapses to one component, to $\beta = 1$. To evaluate the performance of our algorithms combined with such an annealing schedule, we performed the following experiments: mixtures of four t-distributed two-dimensional components with unit covariance were generated with centers picked from a uniform distribution in the $[-5, 5] \times [-5, 5]$ square. A fifth, relatively distant mixture component was added at $(20, 0)$. The mixing proportions of the different components were $(0.1, 0.2, 0.3, 0.3, 0.1)$, respectively, with a total of 1000 data points. 100 random mixtures were generated with each of 6 possible degrees of freedom parameters $\{v = 1, 1.5, 2, 4, 10, 50\}$. Fig. 1a illustrates a typical random mixture ($v = 1$). We fit each mixture with DAEM steps from the first algorithm as well as a standard EM for t-mixtures, initialized at five random points in the $[-20, 20] \times [-20, 20]$ square, with equal mixing proportions. The DAEM algorithm was initialized at $\beta_1 = 0.05$, and run to convergence at each 'temperature', with $\beta_{k+1} = 1.06\,\beta_k$ until $\beta_{final} = 1$. We were able to detect mixture components that reached critical temperatures and split them using a semi-analytical method suggested in Refs. [9,10] and adapted to this problem. To simplify the mixture decomposition problem we set the covariances and degrees of freedom to their known values. For each of the mixtures, the log-likelihoods estimated by the EM and DAEM algorithms were measured. To obtain a rough estimate of the global log-likelihood maximum, we initialized an additional EM algorithm with the true model parameters, and ran it to convergence. Convergence in all cases was assessed as a relative change smaller than 10^{-7} in the log-likelihood. We define a successful estimate to be one with a log-likelihood nearly indistinguishable from that of the presumed global maximum ($\Delta L/L < 10^{-5}$).

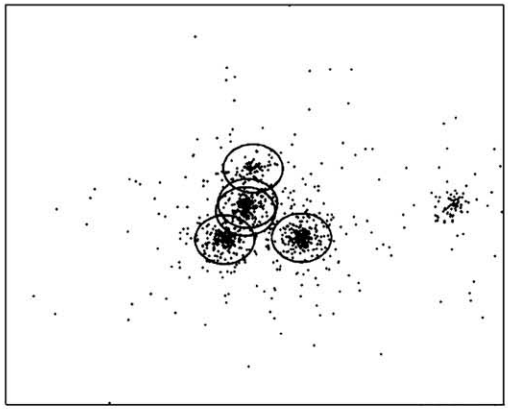
Fig. 1c displays the percentage of successful estimates by the DAEM and EM algorithms as a function of degrees of freedom. While the performance of the DAEM at the near-gaussian limit ($v \gg 1$) is quite remarkable, the performance degrades rapidly for $v < 10$. The transition is not present in the EM algorithm, whose performance deterioration is smoother. A closer look at the DAEM results (Fig. 1b) reveals that the sharp deterioration is caused by an inability of the algorithm to capture the 'distant' component at low DOF. This problem is not mitigated by using EM steps from the second algorithm or by changing the exact annealing schedule or convergence criterion. Although we do not have a full explanation of this phenomenon, it appears that the t-distribution's wider tails down-weigh the distant component and essentially prevent new components from moving away from the high-density region. In contrast, the EM algorithm breakdown was due to components getting stuck in low-density regions, near outlier data points.

3.2. Deterministic agglomeration EM (DAGEM)

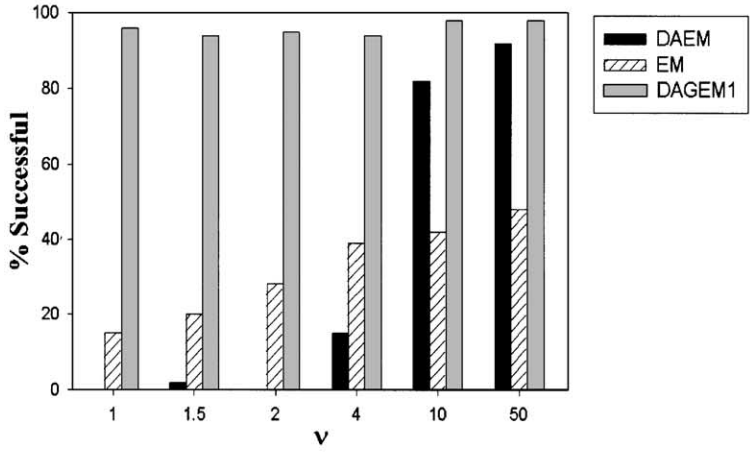
To overcome this problem we turn to a new annealing schedule, agglomerative in nature. We start at $\beta = 1$ with a highly over-specified number of components, and reduce β slowly, going through a monotonically decreasing number of components. Once two or more components converge to the same mean and covariance parameters, we merge them into a single component (with $\pi = \pi_1 + \pi_2$). In addition, as some mixture components will converge towards singularity, we purge components with negligible mixing proportion or whose covariance becomes nearly singular. This step is also effective in removing components that get stuck near



(a)



(b)



(c)

Fig. 1. (a) t-distribution mixture ($v = 1$). The view is restricted to ± 25, with roughly 4% of the points excluded. (b) DAEM result for mixture in (a). (c) Comparative performance of DAEM, EM and DAGEM for mixtures of t-distributions, at different DOF.

outlier data. In the case where we know the exact number of components a priori we stop the agglomeration process when this number is reached and 'cool' the mixture back to $\beta = 1$. More generally, we have to 'cool' models of all different sizes, and compare them at $\beta = 1$ using a penalized likelihood criterion.

Fig. 1 also illustrates the superior performance of DAGEM using EM steps from the first algorithm, with the random mixtures described in the previous section. The estimated mixture was initialized with 15 components placed using a Fuzzy C-Means (FCM) algorithm [32] (with the exponent parameter set to 2). A slower changing annealing schedule ($\beta_{k+1} = \beta_k/1.01$) than that followed by the DAEM was used here. We chose this schedule mainly due to the fact that the 'temperature' range was significantly narrower, with $\beta_{final} \cong 0.6$–0.8 (annealing was stopped when the mixture fell below six components), as well as trying to avoid the practical problem where the number of components jumps between $g > 5$ and < 5. This problem still occurrred in about 0–3% of the mixtures (higher at low DOF), and was resolved by splitting the last merged component during the subsequent 'cooling' phase. We also note that the ratio of components purged/components merged changed from roughly 4 : 1 to 1 : 4 between low and high DOF, respectively, thereby qualitatively changing the algorithm's behavior. We used $\pi_j < 3/N$; $||\boldsymbol{\mu}_i - \boldsymbol{\mu}_j|| < 0.03$ as our purging and merging criteria, respectively.

3.3. Merging criteria

Two or more components whose location and scale parameters become very close are merged during the agglomerative phase. 'Closeness' can be defined using two thresholds: $||\boldsymbol{\mu}_i - \boldsymbol{\mu}_j|| < \varepsilon_1$ and $||\boldsymbol{\Sigma}_i - \boldsymbol{\Sigma}_j|| < \varepsilon_2$, however the required comparisons can become quite cumbersome as they include distance between matrices, and are repeated each EM step. A conceptually simpler approach is to use a distance measure between the component-conditional probabilities of the different data points, which are essentially the $\hat{z}_{ij}$'s. The effect of π_j in these terms should be omitted, as we're only interested in looking at how close location and scale are. Defining: $P_j \equiv \hat{Z}_j/\pi_j$ for component j, we used a criterion based on a normalized distance measure

$$\frac{||P_i - P_j||}{\sqrt{||P_i||||P_j||}} < \delta. \tag{33}$$

In the simulations that follow we used $\delta = 0.01$ as the merging threshold, which corresponded to very similar location and scale. Related, but different, merging criteria were used in Refs. [13,19].

Algorithms

Initialization: use simple clustering method (e.g. k-means or FCM) to determine centers $\boldsymbol{\mu}_{1\ldots g\,\max}$ of $g_{\max} \gg g_{\text{true}}$ components. Set $\pi_{1\ldots g} = \frac{1}{g_{\max}}$; $\boldsymbol{\Sigma}_{1\ldots g} = \boldsymbol{I}$; $\beta = 1$. If v unknown set to $v = 1$.

Repeat:

$\beta = \beta/C(heating)$ *or* $\beta = \beta * C(cooling)$; $C > 1$

Repeat EM steps:

E Step

Update memberships using (12) or (23)

Update weights using (13) or (24)

M step

Update $\pi_{1\ldots g}$ using (14)

Update $\boldsymbol{\mu}_{1\ldots g}$ using (15)

Update $\boldsymbol{\Sigma}_{1\ldots g}$ using (17) or (18) (equal covariances).

(Optional) Update v with (28), using (26) or (29)

Purge nearly singular mixture components

Join components if (30) is true

Until Convergence $\Delta F_\beta/F_\beta < \varepsilon_3$

If $g = g_{\text{final}}$ change from heating to cooling. If number of components is unknown, cool this model and then return and heat. Repeat for all relevant g.

Until $\beta = 1$

4. Simulation experiments

4.1. Unequal covariance, uniform noise, 2-D

Fig. 2 illustrates "snapshots" of our second algorithm at several different 'temperatures', for a contaminated mixture of gaussians with unequal covariances. Uniform noise was added in the ± 20 rectangle, at a 20% level. The mixture was initialized using a 15 component FCM, and $v^* = 1$. Of the mixture components, the 2 on the left are extremely small containing only 3% of the points each. During the cooling phase the estimated v, as well as estimated covariance of the small components increase significantly, largely overshooting their real extent. The reason for the latter effect is that our model, where v is the same for all components, is not suitable for uniform noise, where smaller components effectively have relatively more pronounced tails. In spite of this inadequacy, the algorithm yields a useful answer. An EM algorithm initialized with a 5-component FCM, reaches local log-likelihood maxima on a large proportion of trials.

4.2. High-dimensional, equal covariance, unknown DOF

This experiment was similar to that described in Section 3, however it was repeated in a significantly more general setting. The data were created in 10 dimensions

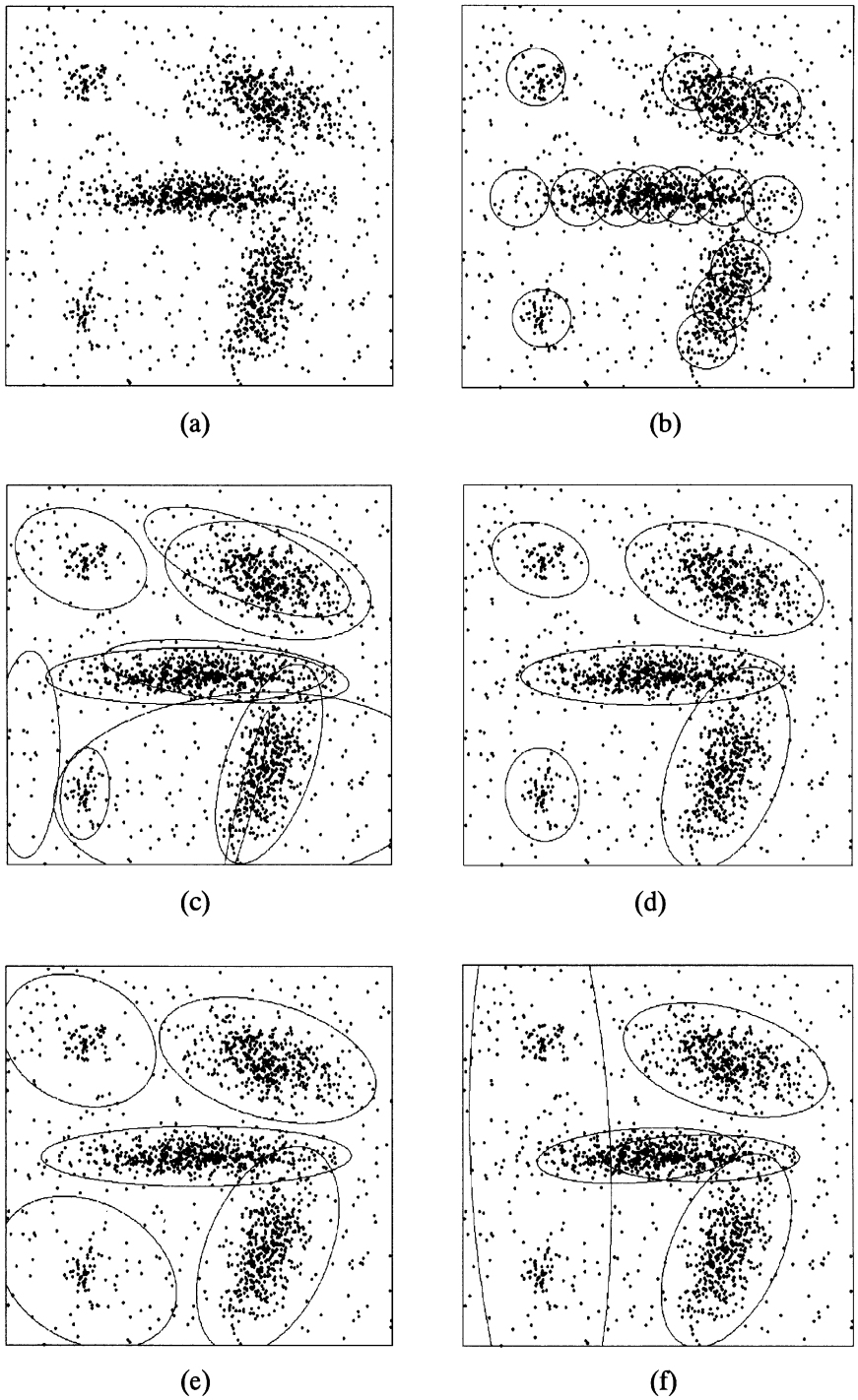


Fig. 2. Results with ellipsoidal clusters and uniform noise (a) Raw data. (b) Initialization. (c) Results at $\beta = 0.83$ ($v^* = 2.7$). (d) Convergence at $\beta = 0.64$ ($v^* = 1.6$). (e) Final result, $\beta = 1$ ($v^* = 3.9$). (f) Spurious solution obtained by EM.

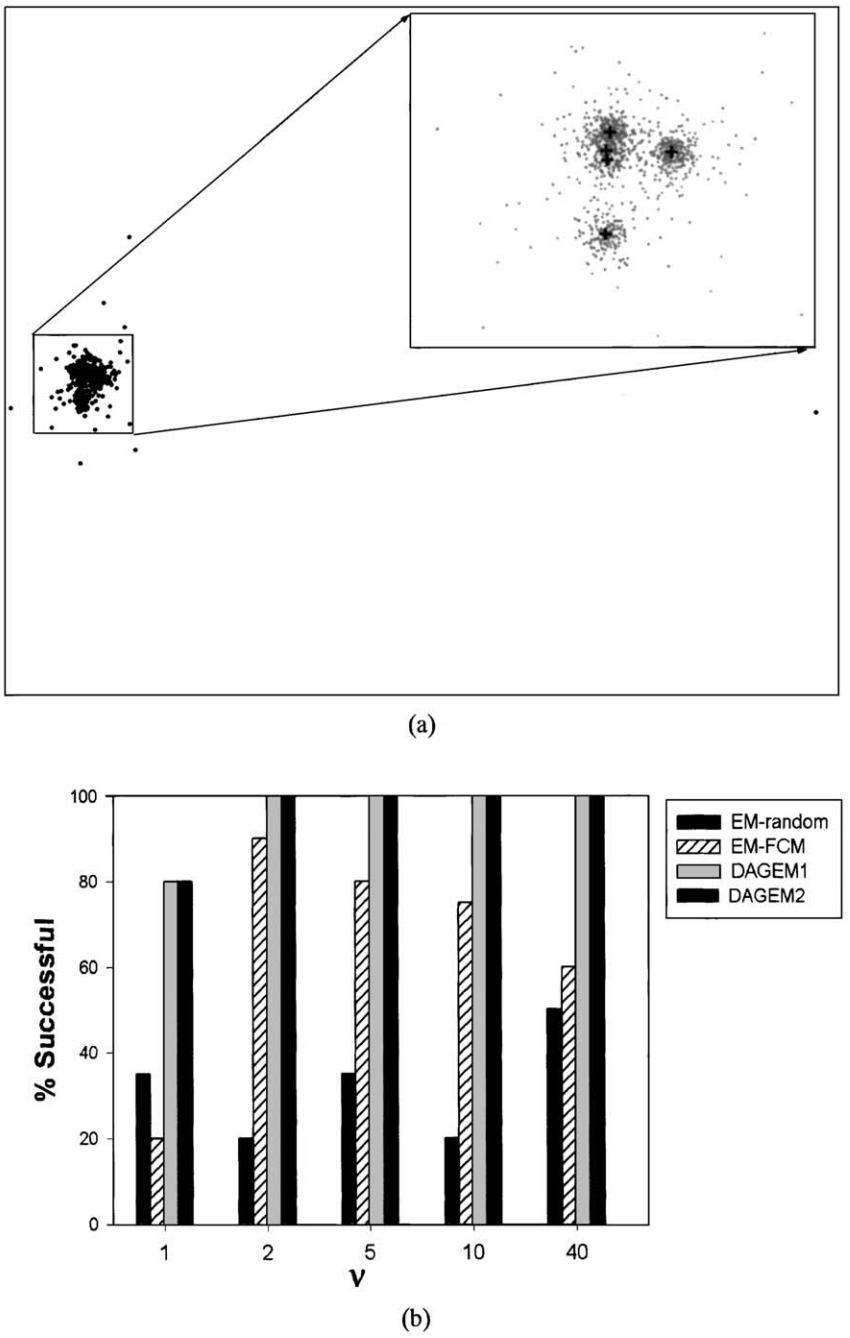


Fig. 3. (a) Scatter plot of a typical ten-dimensional random mixture with five components ($v=2$) projected on first two principal components. Inset is zoomed view of high-density region. Crosses indicate true cluster centers. (b) Results of randomly initialized EM, FCM initialized EM and the two algorithms on mixtures at five different DOF.

with non-isotropic covariance, and both the (shared) covariance and DOF were estimated with the rest of the parameters. The five simulated components had the same mixing proportions as before, their centers were uniformly distributed in the ± 5 hypercube, and they had a diagonal covariance matrix whose elements were uniformly distributed between 0.5 and 3. The covariance was not assumed to be diagonal by the estimation

algorithm, however we did assume equal covariance for all the components. Twenty random mixtures were generated at different contamination ratios $\{v = 1, 2, 5, 10, 40\}$. An illustrative 2-D projection of the data ($v = 2$) is given in Fig. 3a and the simulation results are given in Fig. 3b. We used similar parameters to those in Section 3, except $\beta_{k+1} = \beta_k / 1.03$ (in order to speed up the algorithm). For comparison purposes we ran two EM algorithms, one initialized randomly, and one initialized using an FCM algorithm, which is sometimes used by practitioners as a method of avoiding initialization sensitivity (e.g. Ref. [14]).

5. Discussion

In this paper we develop robust and initialization insensitive algorithms by creating deterministic annealing versions of the EM algorithm for mixtures of multivariate t-distributions. Creating a DAEM framework for this model proved not to be a straightforward extension of DAEM for mixtures of gaussians. As the problem contains two distinct sets of augmented data (binary memberships z_{ij} and typicality weights u_{ij}), two different algorithms are possible, one where both are treated equally (REM2), and a second where they are treated somewhat unequally (Modified DAEM). The latter algorithm was based on a novel modification of DAEM that involves changing the normalizing constraint on the mixing proportion to a 'temperature dependent' constraint.

Our simulation results (Fig. 1) indicate that the DAEM algorithms for this mixture model face considerable difficulty moving components from high-density regions to distant regions, in particular at low degrees of freedom (data sets with many outliers), in contrast to the gaussian case. We resolve this problem by changing to an agglomerative annealing schedule. Our study thus joins a number of recent studies that have looked at improving EM and related algorithms' convergence capability by turning to agglomerative variants [6,14–16]. Previous methods relied on modifying the M step, while our approach, by using the DAEM procedure, is the first to suggest agglomeration driven by a modified E step. The relative merits of the two different approaches to EM based agglomeration (as well as the possibility of hybrid approaches) remain to be addressed by further research. We note that our method requires that the annealing be reversed back to $\beta = 1$ (similar to competitive agglomeration) to obtain the final result. However, our experience with the methods indicates that the 'cooling' phase is extremely rapid relative to the 'heating' one. Thus, if one does not know the optimal number of mixture components, our algorithm, like similar agglomerative algorithms, presents an efficient method of moving between model sizes in an effort to determine model size.

The deterministic agglomeration EM procedure outlined here may prove to be a useful approach in other applications of EM to robust mixtures [9,15,20,26,27]. However, several additional issues with the application need to be addressed. First, as in other applications of maximum likelihood to unconstrained mixtures, we faced the serious problem of mixture components becoming singular. Apart from our ad hoc 'purging' approach, various other approaches may include the Bayesian approach of providing a prior probability on the parameter space [16,18,19,22], or adding various penalties to the likelihood to encourage larger mixture components. One may also be able to tie the regularization to the annealing schedule, as proposed in Ref. [33]. In situations where it is appropriate one can simply restrict the problem to diagonal or equal covariances, which reduces computational requirements and renders the algorithms both faster and more stable. Second, the relative performance advantages of the two algorithms presented (DAGEM1& 2), which achieved identical performance on the simulated problems presented here, remain to be elucidated. A third problematic issue is that of computational load. Roughly speaking, DAGEM is one to two orders of magnitude slower than EM. However, the performance heavily depends on implementation, mainly in terms of covariance constraints, annealing schedule and EM stopping criterion. For example, since we start with a model with a large number of components (whose convergence is much slower), a larger saving in computational time can be obtained by iterating the EM a fixed number of steps, at least until the number of components becomes smaller. The Competitive Agglomeration method [6,14,15] follows this philosophy, allowing only one iteration at each 'temperature', with a dense sampling of different 'temperatures'. Further computational savings can be gained if the annealing schedule is modified to begin with a sparser sampling of temperatures, moving to a denser sampling near phase transitions. Finally, thorough understanding of DAEM's and DAGEM's capability to track and reach global likelihood maxima and 'phase transitions' in DAEM and DAGEM are still lacking, and merit further theoretical investigation. Theoretical analysis may provide new insights into the failure we have described of DAEM for t-distributions, as well as elucidate successful annealing strategies.

Acknowledgements

I wish to thank Professors R.A. Normann, M. Figueiredo, R.D. Nowak and S.S. Nagarajan for valuable input in preparing this manuscript. The work was supported by a State of Utah Center of Excellence contract #95-3365.

Appendix A. derivation of the EM steps

Here, we derive the EM steps (11)–(15) for the model defined by Eqs. (6)–(8), (10)

E step:

Since $Q(\theta')$ in Eq. (10) has a linear dependence on z_{ij} and $z_{ij}u_i$, calculating the expectation is equivalent to replacing z_{ij} and $z_{ij}u_i$ by their expectation with respect to $f(z,u)$. To evaluate these expectations, we first evaluate the missing data probability:

$$f(z_{ij},u_i)=\frac{p(z_{ij},u_i;\theta'^{(k)})\,p(\boldsymbol{x}_i\mid z_{ij},u_i;\theta'^{(k)})}{\int\int p(z_{ij},u_i;\theta'^{(k)})\,p(\boldsymbol{x}_i\mid z_{ij},u_i;\theta'^{(k)})\,\mathrm{d}u_i\,\mathrm{d}z_{ij}}$$
$$=\frac{\pi_j\,p(u_i\mid v)\,p(\boldsymbol{x}_i\mid z_{ij},u_i;\theta'^{(k)})}{\sum_{j=1}^{g}\int_0^\infty \pi_j\,p(u_i\mid v)\,p(\boldsymbol{x}_i\mid z_{ij},u_i;\theta'^{(k)})\,\mathrm{d}u_i}. \tag{A.1}$$

Substituting (6)–(8) and expanding the numerator

$$\pi_j u_i^{v/2-1}\exp\Big(-\frac{v}{2}u_i\Big)\cdot\left[\frac{\exp(-(u_i/2)\cdot\delta(\boldsymbol{x}_i,\boldsymbol{\mu}_j;\boldsymbol{\Sigma}))}{|\boldsymbol{\Sigma}_j|^{1/2}(2\pi/u_i)^{p/2}}\right]$$
$$\propto \pi_j|\boldsymbol{\Sigma}_j|^{-1/2}u_i^{(v+p)/2-1}\exp\left(-u_i\frac{\delta(\boldsymbol{x}_i,\boldsymbol{\mu}_j;\boldsymbol{\Sigma})+v}{2}\right).$$

Then, using the following integral in calculating the expectation:

$$\int_0^\infty u_i^q\exp(-u_i/b)\,\mathrm{d}u_i=\Gamma(a+1)\cdot b^{a+1}. \tag{A.2}$$

We finally obtain

$$\hat{z}_{ij}=E_{f(\boldsymbol{x}_{mis})}(z_{ij})$$
$$=\frac{\pi_j|\boldsymbol{\Sigma}_j|^{-1/2}(\delta(\boldsymbol{x}_i,\boldsymbol{\mu}_j;\boldsymbol{\Sigma}_j)+v)^{-(p+v)/2}}{\sum_{j=1}^{g}\pi_j|\boldsymbol{\Sigma}_j|^{-1/2}(\delta(\boldsymbol{x}_i,\boldsymbol{\mu}_j;\boldsymbol{\Sigma}_j)+v)^{-(p+v)/2}}, \tag{A.3}$$

$$E_{f(\boldsymbol{x}_{mis})}(z_{ij}u_i)=\frac{\pi_j|\boldsymbol{\Sigma}_j|^{-1/2}(\delta(\boldsymbol{x}_i,\boldsymbol{\mu}_j;\boldsymbol{\Sigma}_j)+v)^{-(p+v)/2}((p+v)/(\delta(\boldsymbol{x}_i,\boldsymbol{\mu}_j;\boldsymbol{\Sigma}_j)+v))}{\sum_{j=1}^{g}\pi_j|\boldsymbol{\Sigma}_j|^{-1/2}(\delta(\boldsymbol{x}_i,\boldsymbol{\mu}_j;\boldsymbol{\Sigma}_j)+v)^{-(p+v)/2}}=\hat{z}_{ij}\hat{u}_{ij}, \tag{A.4}$$

where we define $\hat{u}_{ij}$ as

$$\hat{u}_{ij}=\frac{p+v}{\delta(\boldsymbol{x}_i,\boldsymbol{\mu}_j;\boldsymbol{\Sigma}_j)+v}. \tag{A.5}$$

M step:

We have to find the parameters that maximize $Q(\theta)$, under the constraint $\sum_{j=1}^{g}\pi_j=1$.

Mixing proportions: using a Lagrange multiplier λ to enforce the constraint

$$\frac{\mathrm{d}}{\mathrm{d}\pi_j}\left[Q-\lambda\left(\sum_j\pi_j-1\right)\right]=\sum_{i=1}^{N}\frac{z_{ij}}{\pi_j}-\lambda=0,$$

that gives the familiar result:

$$\pi_j=\sum_{i=1}^{N}\frac{\hat{z}_{ij}}{N}. \tag{A.6}$$

Means:

$$\frac{\mathrm{d}}{\mathrm{d}\boldsymbol{\mu}_j}[Q]=\sum_{i=1}^{N}z_{ij}u_{ij}\boldsymbol{\Sigma}_j^{-1}(\boldsymbol{x}_i-\boldsymbol{\mu}_j)=0,$$

which gives

$$\boldsymbol{\mu}_j^{(k)}=\frac{\sum_{i=1}^{N}\hat{z}_{ij}\hat{u}_{ij}\boldsymbol{x}_i}{\sum_{i=1}^{N}\hat{z}_{ij}\hat{u}_{ij}}. \tag{A.7}$$

Covariances:

$$\frac{\mathrm{d}}{\mathrm{d}(\boldsymbol{\Sigma}_j^{-1})}[Q]=-\frac{1}{2}\sum_{i=1}^{N}z_{ij}(\boldsymbol{\Sigma}_j-u_i(\boldsymbol{x}_i-\boldsymbol{\mu}_j^{(k)})(\boldsymbol{x}_i-\boldsymbol{\mu}_j^{(k)})^{\mathrm{T}})$$
$$=0,$$

which gives

$$\boldsymbol{\Sigma}_j^{(k)}=\frac{\sum_{i=1}^{N}(\hat{z}_{ij}\hat{u}_{ij})(\boldsymbol{x}_i-\boldsymbol{\mu}_j^{(k)})(\boldsymbol{x}_i-\boldsymbol{\mu}_j^{(k)})^{\mathrm{T}}}{\sum_{i=1}^{N}\hat{z}_{ij}}. \tag{A.8}$$

Appendix B. REM 2: derivation of the *E* step

It is shown in Ref. [9] that prescribing E and M steps that alternately maximize $F'_\beta(\theta)$ with respect to $f(\boldsymbol{x}_{mis})$ and θ yields

E step: Evaluate $Q'_\beta(\theta)$ in Eq. (19) with

$$f(\boldsymbol{x}_{mis})=\frac{p(\boldsymbol{x}_{mis};\theta^{(k)})\,p(\boldsymbol{x}_{obs}\mid\boldsymbol{x}_{mis};\theta^{(k)})^\beta}{\int p(\boldsymbol{x}_{mis};\theta^{(k)})\,p(\boldsymbol{x}_{obs}\mid\boldsymbol{x}_{mis};\theta^{(k)})^\beta\,\mathrm{d}\boldsymbol{x}_{mis}}. \tag{B.1}$$

M step: maximize $Q'_\beta(\theta)$ with respect to θ.

E step:

$$f(z_{ij},u_i)=\frac{p(z_{ij},u_i;\theta'^{(k)})\,p(\boldsymbol{x}_i\mid z_{ij},u_i;\theta'^{(k)})^{\beta}}{\int\int p(z_{ij},u_i;\theta'^{(k)})\,p(\boldsymbol{x}_i\mid z_{ij},u_i;\theta'^{(k)})^{\beta}\,\mathrm{d}u_i\,\mathrm{d}z_{ij}}$$
$$=\frac{\pi_j\,p(u_i\mid v)\,p(\boldsymbol{x}_i\mid z_{ij},u_i;\theta'^{(k)})^{\beta}}{\sum_{j=1}^{g}\int_0^{\infty}\pi_j\,p(u_i\mid v)\,p(\boldsymbol{x}_i\mid z_{ij},u_i;\theta'^{(k)})^{\beta}\,\mathrm{d}u_i}. \tag{B.2}$$

Substituting (6)–(8) and expanding the numerator:

$$\pi_j u_i^{v/2-1}\exp\Big(-\frac{v}{2}u_i\Big)\cdot\left[\frac{\exp(-(u_i/2)\cdot\delta(\boldsymbol{x}_i,\boldsymbol{\mu}_j;\boldsymbol{\Sigma}))}{|\boldsymbol{\Sigma}_j|^{1/2}(2\pi/u_i)^{p/2}}\right]^{\beta}$$
$$\propto\pi_j|\boldsymbol{\Sigma}_j|^{-\beta/2}u_i^{(v+\beta p)/2-1}\exp\left(-u_i\frac{\beta\delta(\boldsymbol{x}_i,\boldsymbol{\mu}_j;\boldsymbol{\Sigma})+v}{2}\right).$$

Using integral (A.2) we obtain

$$\hat{z}_{ij}=E_{f(\boldsymbol{x}_{mis})}(z_{ij})=\frac{\pi_j|\boldsymbol{\Sigma}_j|^{-\beta/2}(\beta\delta(\boldsymbol{x}_i,\boldsymbol{\mu}_j;\boldsymbol{\Sigma}_j)+v)^{-(\beta p+v)/2}}{\sum_{j=1}^{g}\pi_j|\boldsymbol{\Sigma}_j|^{-\beta/2}(\beta\delta(\boldsymbol{x}_i,\boldsymbol{\mu}_j;\boldsymbol{\Sigma}_j)+v)^{-(\beta p+v)/2}}, \tag{B.3}$$

$$E_{f(\boldsymbol{x}_{mis})}(z_{ij}u_i)=\frac{\pi_j|\boldsymbol{\Sigma}_j|^{-\beta/2}(\beta\delta(\boldsymbol{x}_i,\boldsymbol{\mu}_j;\boldsymbol{\Sigma}_j)+v)^{-(\beta p+v)/2}(\beta p+v)/(\beta\delta(\boldsymbol{x}_i,\boldsymbol{\mu}_j;\boldsymbol{\Sigma}_j)+v)}{\sum_{j=1}^{g}\pi_j|\boldsymbol{\Sigma}_j|^{-\beta/2}(\beta\delta(\boldsymbol{x}_i,\boldsymbol{\mu}_j;\boldsymbol{\Sigma}_j)+v)^{-(\beta p+v)/2}}=\hat{z}_{ij}\hat{u}_{ij}, \tag{B.4}$$

where we define $\hat{u}_{ij}$ as

$$\hat{u}_{ij}\equiv\frac{\beta p+v}{\beta\delta(\boldsymbol{x}_i,\boldsymbol{\mu}_j;\boldsymbol{\Sigma}_j)+v}. \tag{B.5}$$

Appendix C. Modified DAEM: derivation of the EM steps

The *E* and *M* steps for the regular DAEM algorithm are [5]:

E step: Evaluate $Q_\beta(\theta)=E_{f(\boldsymbol{x}_{mis})}[\beta\log(p(\boldsymbol{x}_{obs}\mid\boldsymbol{x}_{mis};\theta)\,p(\boldsymbol{x}_{mis};\theta))]$
with

$$f(\boldsymbol{x}_{mis})=\frac{(p(\boldsymbol{x}_{mis};\theta^{(k)})\,p(\boldsymbol{x}_{obs}\mid\boldsymbol{x}_{mis};\theta^{(k)}))^{\beta}}{\int(p(\boldsymbol{x}_{mis};\theta^{(k)})\,p(\boldsymbol{x}_{obs}\mid\boldsymbol{x}_{mis};\theta^{(k)}))^{\beta}\,\mathrm{d}\boldsymbol{x}_{mis}}. \tag{C.1}$$

M step: maximize $Q_\beta(\theta)$ with respect to θ.

E step:
Repeating the steps that led to Eq. (10), we obtain the following equivalent of the Q function:

$$Q_\beta(\theta')=E_{z,u}\left[\sum_{i=1}^{N}\sum_{j=1}^{g}z_{ij}\left[-\beta\left(\frac{u_i}{2}(\boldsymbol{x}_i-\boldsymbol{\mu}_j)^{\mathrm{T}}\boldsymbol{\Sigma}_j^{-1}(\boldsymbol{x}_i-\boldsymbol{\mu}_j)-\frac{1}{2}\log|\boldsymbol{\Sigma}_j|\right)+\log\pi'_j+C'\right]\right], \tag{C.2}$$

where we have used the new mixing proportion $\pi'_j=\pi_j^{\beta}$.
The missing data probability is

$$f(z_{ij},u_i)=\frac{\pi'_j(p(u_i\mid v)\,p(\boldsymbol{x}\mid z_{ij},u_i;\theta'^{(t)}))^{\beta}}{\sum_{j=1}^{g}\int_0^{\infty}\pi'_j(p(u_i\mid v)\,p(\boldsymbol{x}\mid z_{ij},u_i;\theta'^{(t)}))^{\beta}\,\mathrm{d}u_i}. \tag{C.3}$$

Expanding the denominator, as in Appendices A and B:

$$\left(\pi_j u_i^{(v/2-1)}\exp\Big(-\frac{v}{2}u_i\Big)\,\frac{\exp(-(u_i/2)\cdot\delta(\boldsymbol{x}_i,\boldsymbol{\mu}_j;\boldsymbol{\Sigma})}{|\boldsymbol{\Sigma}_j|^{1/2}(2\pi/u_i)^{p/2}}\right)^{\beta}$$
$$\propto\pi_j^{\beta}|\boldsymbol{\Sigma}_j|^{-\beta/2}u_i^{(\beta(p+v-2+2/\beta)/2)-1}$$
$$\times\exp\left(-u_i\beta\frac{\delta(\boldsymbol{x}_i,\boldsymbol{\mu}_j;\boldsymbol{\Sigma})+v}{2}\right).$$

We easily obtain using integral (A.2) the following expressions for the *E* step updates:

$$\hat{z}_{ij}=\frac{\pi_j^{\beta}|\boldsymbol{\Sigma}_j|^{-\beta/2}(\delta(\boldsymbol{x}_i,\boldsymbol{\mu}_j;\boldsymbol{\Sigma}_j)+v)^{-\beta(p+v-2+2/\beta)/2}}{\sum_{j=1}^{g}\pi_j^{\beta}|\boldsymbol{\Sigma}_j|^{-\beta/2}(\delta(\boldsymbol{x}_i,\boldsymbol{\mu}_j;\boldsymbol{\Sigma}_j)+v)^{-\beta(p+v-2+2/\beta)/2}} \tag{C.4}$$

$$\hat{u}_{ij}=\frac{p+v-2+2/\beta}{\delta(\boldsymbol{x}_i,\boldsymbol{\mu}_j;\boldsymbol{\Sigma})+v}. \tag{C.5}$$

M step:
Maximization of the Q functions in Eqs. (10) and (C.2) result in virtually identical *M* steps. In particular

maximization with respect to the new mixing proportion π'_j, with the usual constraint defined in (24) yields

$$\frac{\mathrm{d}}{\mathrm{d}\pi'_j}\left[Q_\beta - \lambda\left(\sum_j \pi'_j - 1\right)\right] = \beta\left(\sum_{i=1}^{N}\frac{z_{ij}}{\pi'_j} - \lambda\right) = 0.$$

Yields

$$\pi'_j = \frac{\sum_{i=1}^{N}\hat{z}_{ij}}{N}. \tag{C.6}$$

References

[1] G.J. McLachlan, D. Peel, Finite Mixture Models, Wiley, New York, 2000.

[2] A.K. Jain, R.P.W. Duin, J. Mao, Statistical pattern recognition: a review, IEEE Trans. Pattern Anal. Mach. Intell. 22 (1) (2000) 4–37.

[3] A.P. Dempster, N.M. Laird, D.B. Rubin, Maximum likelihood from incomplete data using the EM algorithm (with discussion), J. R. Stat. Soc. B 39 (1977) 1–39.

[4] G.J. McLachlan, T. Krishnan, The EM Algorithm and Extensions, Wiley, New York, 1997.

[5] N. Ueda, R. Nakano, Deterministic annealing EM algorithm, Neural Networks 11 (2) (1998) 271–282.

[6] S. Medasani, R. Krishnapuram, Determination of the number of components in Gaussian mixtures using agglomerative clustering, Proceedings of the 1997 IEEE International Conference on Neural Networks, Houston, Texas, 1997.

[7] K.L. Lange, R.J.A. Little, J.M.G. Taylor, Robust statistical modeling using the t distribution, J. Amer. Stat. Assoc. 84 (408) (1989) 881–896.

[8] D. Peel, G.J. McLachlan, Robust mixture modelling using the t distribution, Stat. Comput. 10 (2000) 339–348.

[9] M. Sahani, Latent Variable Models for Neural Data Analysis, Department of Computation and Neural Systems, California Institute of Technology, Pasadena, California, 1999. http://www.gatsby.ucl.ac.uk./ ∼maneesh/thesis/

[10] K. Rose, Deterministic annealing for clustering, compression, classification, regression, and related optimization problems, Proc. IEEE 86 (11) (1998) 2210–2239.

[11] R.M. Neal, G.E. Hinton, A view of the EM algorithm that justifies incremental, sparse, and other variants, in: M.I. Jordan (Ed.) Learning in Graphical Models, 1998, Kluwer Academic Publishers, Dordrecht, pp. 355–368.

[12] J. Puzicha, T. Hofmann, M.J. Buhmann. Deterministic Annealing: fast physical heuristics for real time optimization of large systems, in: Proceedings of the 15th IMACS World Congress on Scientific Computation, Modeling and Applied Mathematics, Berlin, 1997.

[13] N. Ueda et al., SMEM algorithm for mixture models, Neural Comput. 12 (9) (2000) 2109–2128.

[14] H. Frigui, R. Krishnapuram, Clustering by competitive agglomeration, Pattern Recognition 30 (7) (1997) 1109–1119.

[15] S. Medasani, R. Krishnapuram. Categorization of image databases for efficient retrieval using robust mixture decomposition. IEEE Conference on Computer Vision and Pattern Recognition, IEEE, Santa Barbara, 1998.

[16] M. Figueiredo, A.K. Jain, Unsupervised selection and estimation of finite mixture models, International Conference on Pattern Recognition—ICPR'2000, Barcelona, 2000.

[17] Y. Xinxing, J. Licheng, Fast global optimization fuzzy-neural network and its application in data fusion, Proc. SPIE 3545 (1998) 570–573.

[18] S. Roberts et al., Bayesian approaches to Guassian mixture modeling, IEEE Trans. Pattern Anal. Mach. Intell. 20 (11) (1998) 1133–1142.

[19] M. Figueiredo, J. Leiato, A.K. Jain, On fitting mixture models, in: E. Hancock, M. Pellilo (Eds.), Energy Minimization Methods in Computer Vision and Pattern Recognition, Springer, Berlin, 1999, pp. 45–69.

[20] J.D. Banfield, A.E. Raftery, Model-based Gaussian and non-Gaussian clustering, Biometrics 49 (1993) 803–821.

[21] C. Fraley, Algorithms for model-based Gaussian hierarchical clustering, SIAM J. Sci. Comput. 20 (1) (1998) 270–281.

[22] M. Brand, Structure learning in conditional probability models via entropic prior and parameter extinction, Neural Comput. 11 (1999) 1155–1182.

[23] R.N. Dave, R. Krishnapuram, Robust clustering methods: a unified view, IEEE Trans. Fuzzy Systems 5 (2) (1997) 270–293.

[24] C. Liu, D.B. Rubin, ML estimation of the t distribution using EM and its extensions, ECM and ECME, Stat. Sinica 5 (1) (1995) 19–39.

[25] P.J. Huber, Robust Statistics, Wiley, New York, 1982.

[26] N.A. Campbell, Mixture models and atypical values, Math. Geol. 16 (5) (1984) 465–477.

[27] S. Tadjudin, D.A. Landgrebe, Robust parameter estimation for mixture model, IEEE Trans. Geosic. Remote Sens. 38 (1) (2000) 439–445.

[28] X.L. Meng, D.A. van Dyk, The EM algorithm—an old folk-song sung to a fast new tune, J. R. Stat. Soc. Ser. B 59 (3) (1997) 511–567.

[29] J.T. Kent, D.E. Tyler, Y. Vardi, A curious likelihood identity for the multivariate t-distribution, Commun. Stat. Simulation Comput. 23 (2) (1994) 441–453.

[30] C. Liu, ML estimation of the multivariate t distribution and the EM Algorithm, J. Multivar. Anal. 63 (2) (1997) 296–312.

[31] C. Liu, D.B. Rubin, Y. Wu, Parameter expansion to accelerate EM: the PX-EM algorithm, Biometrika 85 (4) (1998) 755–770.

[32] J.C. Bezdek, Pattern recognition with fuzzy objective function algorithms, Plenum Press, New York, 1981.

[33] M. Kloppenburg, P. Tavan, Deterministic annealing for density estimation by multivariate normal mixtures, Phys. Rev. E 55 (3) (1996) R2089–R2092.

About the Author—SHY SHOHAM received his B.Sc. in Physics in 1993 from Tel Aviv University, Israel. Since 1996, he is pursuing a Ph.D. in Bioengineering at the University of Utah. His work at the Center for Neural Interfaces involves developing robust and efficient clustering and estimation algorithms for an implantable Brain–Computer interface based on multi-electrode arrays. His general interests include Applied Neurophysiology, Brain Imaging, Computational Neuroscience and Statistical Signal Processing.