Computing, Artificial Intelligence and Information Technology

Estimation of parameters in latent class models using fuzzy clustering algorithms

Miin-Shen Yang *, Nan-Yi Yu

Department of Applied Mathematics, Chung Yuan Christian University, Chung-Li 32023, Taiwan, ROC

Received 18 March 2002; accepted 30 July 2003
Available online 14 November 2003

Abstract

A mixture approach to clustering is an important technique in cluster analysis. A mixture of multivariate multinomial distributions is usually used to analyze categorical data with latent class model. The parameter estimation is an important step for a mixture distribution. Described here are four approaches to estimating the parameters of a mixture of multivariate multinomial distributions. The first approach is an extended maximum likelihood (ML) method. The second approach is based on the well-known expectation maximization (EM) algorithm. The third approach is the classification maximum likelihood (CML) algorithm. In this paper, we propose a new approach using the so-called fuzzy class model and then create the fuzzy classification maximum likelihood (FCML) approach for categorical data. The accuracy, robustness and effectiveness of these four types of algorithms for estimating the parameters of multivariate binomial mixtures are compared using real empirical data and samples drawn from the multivariate binomial mixtures of two classes. The results show that the proposed FCML algorithm presents better accuracy, robustness and effectiveness. Overall, the FCML algorithm has the superiority over the ML, EM and CML algorithms. Thus, we recommend FCML as another good tool for estimating the parameters of mixture multivariate multinomial models. © 2003 Elsevier B.V. All rights reserved.

Keywords: Categorical data; Latent class model; Fuzzy class model; Maximum likelihood function; Fuzzy clustering algorithms

1. Introduction

Since the latent class model was first proposed by Green [9] and Lazarsfeld [13], it has been widely applied in many areas of psychology, sociology and economics, etc. (see Everitt [5], Ingledew et al. [10], Rees et al. [18]). The parameter estimation is an important step in the latent class model. In a number of procedures for estimating the parameters of latent class models, the EM algorithm is effectively used to find the maximum likelihood estimates when latent class analysis is formulated as a problem of estimating parameters in a mixture of distributions (see Everitt [6], Galecki et al. [7]). The latent class model is effectively used as the analysis of grouped categorical data. However, the latent class variables are assumed to
be crisp that present the probabilities of memberships of objects to each class by reflecting the memberships of objects with uncertainty.

Since Zadeh [27] proposed fuzzy set theory which produced the idea of partial membership described by a membership function, fuzziness has received more attention. Fuzziness is a different concept from stochastic (or probability). Kosko [11] and Laviolette et al. [12] gave good discusses for fuzziness and probability. Fuzzy clustering which presents overlapping cluster partitions has been widely studied and applied in various areas (see Bezdek [1], Wu and Yang [22], Yang [25]). Similarly, the use of fuzzy sets can provide a method for extending latent class models into fuzzy class models. This allows embedding fuzzy clustering methods into fuzzy class models for categorical data to include partial (or overlapping clusters) memberships.

Mixtures of distributions are used as models in statistical studies and also have applications to clustering (see Mclachlan and Basford [16]). A mixture of multivariate multinomial distributions is used to analyze grouped categorical data. Described here are four approaches for estimating the parameters of a mixture of multivariate multinomial distributions. The first approach is an extended maximum likelihood (ML) method originally proposed by Goodman [8]. The second approach is based on the well-known expectation maximization (EM) algorithm (see Dempster et al. [4], McLachlan and Krishnan [17]). The third approach is the classification maximum likelihood (CML) algorithm discussed by Celeux and Govaert [3]. In this paper, we propose a new approach using the so-called fuzzy class model and then create the fuzzy classification maximum likelihood (FCML) approach for categorical data. The accuracy and effectiveness of these four types of algorithms for estimating the parameters of the multivariate binomial mixtures are compared using real empirical data and samples drawn from multivariate binomial mixtures of two classes.

In Section 2, we describe the latent class model and derive corresponding algorithms such as the ML, EM and CML algorithms. In Section 3, we extend the latent class model into the fuzzy class model and derive its corresponding algorithm called the fuzzy CML (FCML) algorithm. Section 4 gives examples and numerical comparisons. Conclusions will be made in Section 5.

2. Latent class model and its algorithms

Let $X = \{x_1, \ldots, x_n\}$ be a set of grouped categorical data with $x_i = x_{ih}^j$ by $q$ categorical variables, with respective number of categories $m_1, \ldots, m_q$ for $m = \sum_{j=1}^q m_j$ the total number of categories where $x_{ih}^j; h = 1, \ldots, m_j; j = 1, \ldots, q; i = 1, \ldots, n$ are $x_{ih}^j = 1$ if the datum $x_i$ belongs to the category $h$ of the variable $j$ and 0 otherwise.

In general, the latent class model is used to analyze grouped categorical data. The latent class model assumes that the random observations follow a distribution which is a mixture of multivariate multinomial distributions given by

$$f(x; \alpha, \theta) = \sum_{k=1}^c \alpha_k f_k(x; \theta_k)$$

with

$$f_k(x; \theta_k) = \prod_{j=1}^q \prod_{h=1}^{m_j} (\theta_{kh}^j)^{x_{ih}^j}, \quad k = 1, \ldots, c,$$

(1)

where $\theta_{kh}^j$ gives the probability of category $h$ of the variable $j$ in the class $k$ with $\sum_{h=1}^{m_j} \theta_{kh}^j = 1$ for $j = 1, \ldots, q$ and $k = 1, \ldots, c$, and $\sum_{k=1}^c \alpha_k = 1$ for $\alpha_k > 0$, $k = 1, \ldots, c$.

One choice for estimating parameters in the latent class model (1) is to use the maximum likelihood (ML) principle. The ML principle for estimating the parameters of a mixture of multivariate multinomial
distributions is an iterative optimization procedure known as the ML algorithm. This algorithm for a mixture of multivariate Bernoulli distributions had been derived and applied in Goodman [8]. A brief discussion on the extended ML algorithm to a mixture of multivariate multinomial distributions is given as follows.

Let \( X = \{x_1, \ldots, x_n\} \) be a random sample of categorical data and let \( c \) be a positive integer greater than 1. In accordance with the mixture density \( f(x; \theta) \) of the multivariate multinomial distributions, the log-likelihood for the dataset is given by

\[
L(x, \theta; X) = \sum_{i=1}^{n} \ln \left( \sum_{k=1}^{c} x_k \prod_{j=1}^{q} \prod_{h=1}^{m_j} (\theta_k^{jh})^{x_{jh}} \right),
\]

where \( \theta_k^{jh} \) gives the probability of category \( h \) of the variable \( j \) in the class \( k \) with \( \sum_{h=1}^{m_j} \theta_k^{jh} = 1 \) for \( j = 1, \ldots, q \) and \( k = 1, \ldots, c \), and \( \sum_{k=1}^{c} x_k = 1 \) for \( x_k > 0, k = 1, \ldots, c \).

Similar to the Goodman’s ML method [8] for a mixture of multivariate Bernoulli distributions, we can derive the ML update equations for maximizing the log likelihood (2). Because the mixing proportion \( x_k \) and the probability \( \theta_k^{jh} \) of category \( h \) have the restrictions with \( \sum_{k=1}^{c} x_k = 1 \) and \( \sum_{h=1}^{m_j} \theta_k^{jh} = 1 \), we need to consider the Lagrangian \( \tilde{L} \) with

\[
\tilde{L}(x, \theta, \lambda) = \sum_{i=1}^{n} \ln \left( \sum_{k=1}^{c} x_k \prod_{j=1}^{q} \prod_{h=1}^{m_j} (\theta_k^{jh})^{x_{jh}} \right) - \lambda_1 \left( \sum_{k=1}^{c} x_k - 1 \right) - \lambda_2 \left( \sum_{h=1}^{m_j} \theta_k^{jh} - 1 \right).
\]

We take the partial derivatives of \( \tilde{L}(x, \theta, \lambda) \) with respect to all parameters and set it to be zero. We have

\[
\frac{\partial}{\partial x_k} \tilde{L} = \sum_{i=1}^{n} \frac{\prod_{j=1}^{q} \prod_{h=1}^{m_j} (\theta_k^{jh})^{x_{jh}}}{\sum_{x=1}^{c} \prod_{j=1}^{q} \prod_{h=1}^{m_j} (\theta_x^{jh})^{x_{jh}}} - \lambda_1 = 0, \tag{3}
\]

\[
\frac{\partial}{\partial \theta_k^{jh}} \tilde{L} = \sum_{i=1}^{n} \frac{x_{jh} \prod_{l=1}^{q} \prod_{j=1}^{m_j} (\theta_l^{j})^{x_{jl}}}{\sum_{x=1}^{c} \prod_{j=1}^{q} \prod_{h=1}^{m_j} (\theta_x^{jh})^{x_{jh}}} - \theta_k^{jh} \lambda_2 = 0, \tag{4}
\]

\[
\frac{\partial}{\partial \lambda_1} \tilde{L} = - \left( \sum_{k=1}^{c} x_k - 1 \right) = 0, \tag{5}
\]

\[
\frac{\partial}{\partial \lambda_2} \tilde{L} = - \left( \sum_{h=1}^{m_j} \theta_k^{jh} - 1 \right) = 0. \tag{6}
\]

By solving Eqs. (3) and (5), we have

\[
\sum_{i=1}^{n} \frac{\prod_{j=1}^{q} \prod_{h=1}^{m_j} (\theta_k^{jh})^{x_{jh}}}{\sum_{x=1}^{c} \prod_{j=1}^{q} \prod_{h=1}^{m_j} (\theta_x^{jh})^{x_{jh}}} - n = 0. \tag{7}
\]

We multiply \( x_k \) to both sides of (7). We have

\[
x_k = \frac{1}{n} \sum_{i=1}^{n} \frac{x_k \prod_{j=1}^{q} \prod_{h=1}^{m_j} (\theta_k^{jh})^{x_{jh}}}{\sum_{x=1}^{c} \prod_{j=1}^{q} \prod_{h=1}^{m_j} (\theta_x^{jh})^{x_{jh}}}.
\]

Note that the \( x_k \) in Eq. (8) cannot be directly solved. However, we can use the fixed-point iterative method to approximate it. Let the right-hand side of Eq. (8) be assigned to \( T(x) \). We first specify the initial value \( x^{(0)} = (x_1^{(0)}, \ldots, x_c^{(0)}) \) and compute \( T(x^{(0)}) \). We then assign it to \( x_k^{(1)} \), for \( k = 1, \ldots, c \), according to Eq. (8).
Repeat the step until the \((s+1)\)th solution \(z_k^{(s+1)}\) is very close to the \(s\)th solution. Thus, we may write the update equation as

\[
\hat{z}_k = \frac{1}{n} \sum_{i=1}^{n} \frac{\alpha_k \prod_{i=1}^{m_i} (\theta_k^{(s)})^{x_{ki}} \prod_{j=1}^{q} (\phi_k^{(s)})^{z_{kj}}}{\sum_{c=1}^{c} \alpha_c \prod_{i=1}^{m_i} (\theta_c^{(s)})^{x_{ci}}}, \quad k = 1, \ldots, c. \tag{9}
\]

Similarly, by solving Eqs. (4) and (6) with the fact that \(\sum_{h=1}^{m_j} x_{jh} = 1\), we can derive the fixed-point iteration with the following update equation:

\[
\hat{\theta}_k^{(s)} = \frac{1}{n z_k} \sum_{i=1}^{n} \frac{\alpha_k x_{ki} \prod_{i=1}^{m_i} (\theta_k^{(s)})^{x_{ki}} \prod_{j=1}^{q} (\phi_k^{(s)})^{z_{kj}}}{\sum_{c=1}^{c} \alpha_c \prod_{i=1}^{m_i} (\theta_c^{(s)})^{x_{ci}}}, \quad h = 1, \ldots, m_j, \ j = 1, \ldots, q, \ k = 1, \ldots, c. \tag{10}
\]

Based on updating Eqs. (9) and (10), we give the following ML algorithm.

**ML Algorithm**

\textbf{Step 1:} Fix \(2 \leq c \leq n\) and fix any \(\epsilon > 0\).

Give initial values \(z_k^{(0)}\) and \(\theta_k^{(0)}\) and let \(s = 1\).

\textbf{Step 2:} Compute \(\hat{z}_k^{(s)} = (\hat{z}_k^{(s)}, \hat{\theta}_k^{(s)})\) with \(\gamma^{(s-1)} = (\hat{z}_k^{(s-1)}, \hat{\theta}_k^{(s-1)})\) by (9) and (10).

\textbf{Step 3:} Compare \(\hat{\gamma}^{(s)}\) to \(\gamma^{(s-1)}\).

\textbf{IF} \(\|\hat{\gamma}^{(s)} - \gamma^{(s-1)}\| < \epsilon\), \textbf{STOP}

\textbf{ELSE} \(s = s + 1\), return to \textbf{Step 2}.

In estimating the parameters of latent class models, the EM algorithm had been used as an effective method for approximating maximum likelihood estimates (e.g. Celeux and Govaert [3], Everitt [6], Galecki et al. [7]). Next, we shall give its derivation. Consider \(X = \{x_1, \ldots, x_n\}\) to be an incomplete dataset and latent class variables \(z_1, \ldots, z_c\) to be missing where \(z_{ki} = z_k(x_i) = 1\) if \(x_i\) belongs to \(k\)th class and 0 otherwise for \(k = 1, \ldots, c\) and \(i = 1, \ldots, n\). The EM algorithm is applied to the mixture distributions by treating \(z\) as missing data. The algorithm is easy to program in two steps, \(E\) (for expectation) and \(M\) (for maximization) (see Dempster et al. [4]).

To derive the EM algorithm, we consider \(X = \{x_1, \ldots, x_n\}\) as an incomplete dataset. Let \(z = (z_1, \ldots, z_c)\) or \(z' = (z_1, \ldots, z_c)\) with \(z_{ki} = z_k(x_i) = 1\) and \(z_{k} = (z_{k1}, \ldots, z_{kn})'\) and \(z_i = (z_{i1}, \ldots, z_{ic})'\) for \(k = 1, \ldots, c\) and \(i = 1, \ldots, n\). The latent class variable \(z' = (z_1, \ldots, z_c)\) (or \(z = (z_1, \ldots, z_c)\)) is considered to be missing with its distribution function \(f(z) = \prod_{k=1}^{c} \alpha_k^{z_k}\) where \(\alpha_k\) is the mixing proportion with \(\sum_{k=1}^{c} \alpha_k = 1\). Thus the distribution of \(x_i\) given \(z_i\) is

\[f(x_i|z_i) = \prod_{k=1}^{c} (f_k(x_i; \theta_k))^{z_{ki}}\]

and the joint distribution of \(x\) and \(z\) shall be

\[f(x_1, \ldots, x_n, z_1, \ldots, z_n) = \prod_{i=1}^{n} \left( \prod_{k=1}^{c} (f_k(x_i; \theta_k))^{z_{ki}} \alpha_k \right)\]

Thus, the log likelihood for the complete data is given by

\[L_{\text{EM}}(\alpha; \theta; x_1, \ldots, x_n, z_1, \ldots, z_n) = \sum_{i=1}^{n} \sum_{k=1}^{c} z_{ki} \{\ln \alpha_k f_k(x_i; \theta_k)\},\]
where the density \( f_k(x_i; \theta_k) = \prod_{h=1}^{m_j} (\theta_k^{j_h})^{x_{ki}} \) for \( k = 1, \ldots, c, i = 1, \ldots, n \) is considered. In \( E \)-step, because all \( z_i \) are missing data, we use the expectation \( E(z_i|x_i) \) to estimate the \( z_{ki} \). Because

\[
f(z_{ki}|x_i) = \frac{f(x_i, z_{ki})}{f(x_i)} = \frac{f(x_i|z_{ki})f(z_{ki})}{\sum_{i=1}^{n} f(x_i|z_{ki})f(z_{ki})} = \frac{(f_k(x_i; \theta_k))^{z_{ki}} \pi_k^{z_{ki}}}{\sum_{i=1}^{n} (f_k(x_i; \theta_k))^{z_{ki}}},
\]

\[
\hat{z}_{ki} = E(z_{ki}|x_i) = \frac{\pi_k f_k(x_i; \theta_k)}{\sum_{i=1}^{n} \pi_k f_k(x_i; \theta_k)} + 0 \cdot \frac{1 - 1}{\sum_{i=1}^{n} \pi_k f_k(x_i; \theta_k)} = \frac{\pi_k f_k(x_i; \theta_k)}{\sum_{k=1}^{c} \pi_k f_k(x_i; \theta_k)}.
\]

We now have the expectation \( E(L_{EM}) \) of log likelihood \( L_{EM}(x, \theta; x, z) \) of the complete data with

\[
E(L_{EM}) = \sum_{i=1}^{n} \sum_{k=1}^{c} \hat{z}_{ki} \ln(\pi_k f_k(x_i; \theta_k)).
\]

In \( M \)-step, we maximize \( E(L_{EM}) \) with the restriction that \( \sum_{k=1}^{c} \pi_k = 1 \). That is, we should consider the following Lagrangian:

\[
\tilde{L}_{EM}(x, \theta, \lambda) = \sum_{i=1}^{n} \sum_{k=1}^{c} \hat{z}_{ki} \ln(\pi_k f_k(x_i; \theta_k)) - \lambda \left( \sum_{k=1}^{c} \pi_k - 1 \right).
\]

One can take the first derivatives of \( \tilde{L}_{EM}(x, \theta, \lambda) \) with respect to all parameters and set it to be zero. Thus, the necessary conditions for the maximizer of \( E(L_{EM}) \) can be derived as follows:

\[
\pi_k = \frac{\sum_{i=1}^{n} \hat{z}_{ki} / n}{k = 1, \ldots, c}, \quad (11)
\]

\[
\theta_k^{j_h} = \frac{\sum_{i=1}^{n} \hat{z}_{ki} x_{ki}^{j_h}}{\sum_{i=1}^{n} \hat{z}_{ki}}, \quad k = 1, \ldots, c, \quad j = 1, \ldots, q, \quad h = 1, \ldots, m_j, \quad (12)
\]

in which

\[
z_{ki} = z_k(x_i) = \frac{\pi_k f_k(x_i; \theta_k)}{\sum_{i=1}^{n} \pi_k f_k(x_i; \theta_k)}, \quad k = 1, \ldots, c, \quad i = 1, \ldots, n. \quad (13)
\]

In fact, Eq. (13) shows that \( z_k(x_i) \) may be interpreted via Bayes’ rule as the posterior probability that given \( x_i \), it is drawn from class \( k \), i.e.

\[
z_k(x_i) = \text{Pr}(\text{class } k|x_i; \theta_k).
\]

The \( E \) step is obtained using Eq. (13) and the \( M \) step is used to estimate the parameters \( \pi \) and \( \theta \) by means of (11) and (12). We can repeatedly alternate the \( E \) and \( M \) steps. They have a sequence of executions for stage \( s \) using stage \( s - 1 \). Based on the necessary conditions (11)–(13), we have the EM algorithm as follows:

**EM algorithm**

**Step 1**: Fix \( 2 \leq c \leq n \) and fix any \( \varepsilon > 0 \).

  - Give initial latent class \( z^{(0)} \) and let \( s = 1 \).

**Step 2**: Compute \( \gamma^{(s)} = (\pi^{(s)}, \theta^{(s)}) \) with \( z^{(s-1)} \) by (11) and (12).

**Step 3**: Update to \( z^{(s)} \) with \( \gamma^{(s)} = (\pi^{(s)}, \theta^{(s)}) \) by (13).

**Step 4**: Compare \( z^{(s)} \) to \( z^{(s-1)} \) in a convenient matrix norm.

  - IF \( \|z^{(s)} - z^{(s-1)}\| < \varepsilon \), STOP
  - ELSE \( s = s + 1 \), return to step 2.
The classification maximum likelihood (CML) is a classification approach to approximate the maximum likelihood estimates (see McLachlan and Basford [16]). Let \( X = \{x_1, \ldots, x_n\} \) be a random sample of size \( n \) from a mixture of multivariate multinomial distributions \( f(x; \mathbf{z}, \theta) \) of (1). Let \( P = (P_1, \ldots, P_c) \) be a partition on \( X \). Then, the joint density of \( X \) is \( \prod_{i=1}^{c} \prod_{x \in P_i} z_kf_k(x; \theta_k) \). In the CML method, \( P = (P_1, \ldots, P_c) \), \( \mathbf{z} = (z_1, \ldots, z_c) \) and \( \theta = (\theta_1, \ldots, \theta_c) \) are estimated by maximizing the log likelihood function \( \sum_{i=1}^{n} \sum_{x \in P_i} \ln z_kf_k(x; \theta_k) \). We then have the CML objective function for grouped categorical data \( \mathbf{L}_{\text{CML}} \) with

\[
\mathbf{L}_{\text{CML}}(P, \mathbf{z}, \theta; X) = \sum_{k=1}^{c} \ln z_kf_k(x; \theta_k) = \sum_{k=1}^{c} \sum_{x \in P_k} \ln z_k + \sum_{j=1}^{q} \sum_{h=1}^{m_j} x_{jh} \ln \theta_{kh}^j
\]

subject to \( \sum_{h=1}^{m_j} \theta_{kh}^j = 1 \) and \( \sum_{k=1}^{c} \alpha_k = 1 \), \( \alpha_k > 0 \) for \( j = 1, \ldots, q \) and \( k = 1, \ldots, c \).

For a given partition \( P = (P_1, \ldots, P_c) \), we take the first derivatives of the Lagrangian

\[
\tilde{\mathbf{L}}_{\text{CML}}(P, \mathbf{z}, \theta, \lambda_1, \lambda_2; X) = \mathbf{L}_{\text{CML}}(P, \mathbf{z}, \theta; X) - \lambda_1 \left( \sum_{k=1}^{c} \alpha_k - 1 \right) - \lambda_2 \left( \sum_{h=1}^{m_j} \theta_{kh}^j - 1 \right)
\]

with respect to \( \alpha_k, \theta_{kh}^j, \lambda_1 \) and \( \lambda_2 \) and set them to be 0. We can get the necessary conditions of a maximizer of \( \mathbf{L}_{\text{CML}}(P, \mathbf{z}, \theta; X) \) as follows:

\[
\alpha_k = \frac{|P_k|}{n}, \quad k = 1, \ldots, c,
\]

(14)

and

\[
\theta_{kh}^j = \frac{\sum_{x \in P_k} x_{jh}^j}{|P_k|}, \quad k = 1, \ldots, c, \quad j = 1, \ldots, q, \quad h = 1, \ldots, m_j,
\]

(15)

where \( |P_k| \) is the cardinality of \( P_k \). We then have an alternating optimization algorithm for maximizing \( \mathbf{L}_{\text{CML}}(P, \mathbf{z}, \theta; X) \) as follows:

**CML Algorithm**

*Step 1:* Give an initial partition \( P^{(0)} \) and let \( s = 1 \).

*Step 2:* Find \( \mathbf{z}^{(s)} \) using Eq. (14).

*Step 3:* Find \( \theta^{(s)} \) using Eq. (15).

*Step 4:* Find \( P^{(s)} = \arg \min_p \mathbf{L}_{\text{CML}}(P, \mathbf{z}^{(s)}, \theta^{(s)}). \)

IF \( P^{(s)} = P^{(s-1)} \), STOP

ELSE \( s = s + 1 \), return Step 2.

Note that the special case of the CML algorithm with \( \alpha_k = \frac{1}{c} \) and also with the multivariate binomial distribution was used in numerical comparisons by Celeux and Govaert [3]. The CML algorithm will be extended to the fuzzy CML algorithm with a fuzzy class model in the next section.

### 3. Fuzzy class model and its algorithms

The description of EM and CML objective functions in Section 2 used the latent class variables \( z_1, \ldots, z_c \) and a partition \( P = (P_1, \ldots, P_c) \) on a dataset \( X \). In fact, a partition \( P = (P_1, \ldots, P_c) \) of \( X \) into \( c \) parts can be represented using mutually disjoint sets \( P_1, \ldots, P_c \) such that \( P_1 \cup \cdots \cup P_c = X \) or equivalently using the indicator functions \( \mu_1, \ldots, \mu_c \) such that \( \mu_k(x) = 1 \) if \( x \in P_k \) and \( \mu_k(x) = 0 \) if \( x \notin P_k \) for all \( x \) in \( X \) and for all \( k = 1, \ldots, c \). This is known as clustering \( X \) into \( c \) classes using \( \mu = (\mu_1, \ldots, \mu_c) \) and called a hard \( c \)-partition of \( X \). We can see that the hard \( c \)-partition \( \mu_1, \ldots, \mu_c \) are exactly the same as latent class variables \( z_1, \ldots, z_c \). Thus the CML objective function becomes
where \( P = (P_1, \ldots, P_c) \) is a partition of \( X \) and \( \mu = (\mu_1, \ldots, \mu_c) \) is its hard \( c \)-partition. The CML algorithm is an optimization procedure by choosing a hard \( c \)-partitions \( \mu^* \) and a proportion \( \alpha^* \) and an estimate \( \theta^* \) to maximize the log likelihood \( L_{\text{CML}}(\mu, \alpha, \theta; X) \).

Consider the extension to allow the indicator functions \( \mu_k(x_i) \) to be functions (known as membership functions) assuming values in the interval [0, 1] such that \( \sum_{k=1}^{c} \mu_k(x_i) = 1 \) for all \( x_i \in X \). In this case, the data set \( X \) is said to have a fuzzy \( c \)-partition \( \mu = (\mu_1, \ldots, \mu_c) \) (see Bezdek [1], Yang [25]) and the model is called a fuzzy class model according to basic idea of fuzzy sets proposed by Zadeh [27]. Now the CML procedure becomes

\[
\text{Maximize} \quad J_1(\mu, \alpha, \theta; X) = \sum_{k=1}^{c} \sum_{i=1}^{n} \mu_k(x_i) \ln f_k(x_i; \theta) + \sum_{k=1}^{c} \sum_{i=1}^{n} (1-\alpha_k) \mu_k(x_i) \ln \alpha_k
\]

subject to \( \sum_{k=1}^{c} \alpha_k = 1, \alpha_k > 0 \) and \( \sum_{k=1}^{c} \mu_k(x_i) = 1 \) for \( i = 1, \ldots, n \) with \( \mu_k(x_i) \in [0, 1] \).

According to the optimization problems in (16) and (17), the only difference is that \( \mu_k(x_i) \in \{0, 1\} \) in (16) but \( \mu_k(x_i) \in [0, 1] \) in (17). Since the objective functions are linear in \( \mu_k(x_i) \), the optimal solutions occur at the end point 0 or 1 in both (16) and (17). Therefore, the optimization problem in (17) should be equivalent to that in (16). If we want to make fuzzy extension of the optimization in \( J_1(\mu, \alpha, \theta; X) \), it is necessary to increase the power of \( \mu_k(x_i) \) to \( \mu_k^m(x_i) \), for which \( m > 1 \) represents the degree of fuzziness. Thus the fuzzy extension of the CML procedure may be created by maximizing the following fuzzy CML (FCML) objective function:

\[
J_{\text{FCML}}(\mu, \alpha, \theta; X) = \sum_{k=1}^{c} \sum_{i=1}^{n} \mu_k^m \ln f_k(x_i; \theta) + \sum_{k=1}^{c} \sum_{i=1}^{n} (1-\alpha_k) \mu_k^m \ln \alpha_k
\]

subject to \( \sum_{k=1}^{c} \alpha_k = 1, \alpha_k > 0 \) and \( \sum_{k=1}^{c} \mu_k^m(x_i) = 1 \) for \( i = 1, \ldots, n \) with \( \mu_k^m(x_i) \in [0, 1] \).

Let us now consider the distribution \( f_k(x; \theta_k) \) to be a multivariate multinomial distribution. Then

\[
J_{\text{FCML}}(\mu, \alpha, \theta; X) = \sum_{k=1}^{c} \sum_{i=1}^{n} \mu_k^m \ln f_k(x_i; \theta) + \sum_{k=1}^{c} \sum_{i=1}^{n} (1-\alpha_k) \mu_k^m \ln \alpha_k
\]

subject to \( \sum_{k=1}^{c} \alpha_k = 1, \alpha_k > 0 \) and \( \sum_{k=1}^{c} \mu_k^m(x_i) = 1 \) for \( i = 1, \ldots, n \) with \( \mu_k^m(x_i) \in [0, 1] \).

The optimization problem in \( J_{\text{FCML}}(\mu, \alpha, \theta; X) \) by choosing a fuzzy \( c \)-partition \( \mu \) and a proportion \( \alpha \) and an estimate \( \theta \) to maximize \( J_{\text{FCML}}(\mu, \alpha, \theta; X) \) was called a class of FCML procedures proposed by Yang [24]. If the density \( f_k(x; \theta) \) in \( J_{\text{FCML}} \) is a multivariate standard normal distribution, Yang [24] had derived a penalized fuzzy c-means (PFCM) algorithm which is a generalized type of the well-known FCM algorithm. These PFCM had been applied in segmentation and vector quantization (see [14,15]).

Let us now consider the distribution \( f_k(x; \theta_k) \) to be a multivariate multinomial distribution. Then

\[
J_{\text{FCML}}(\mu, \alpha, \theta; X) = \sum_{k=1}^{c} \sum_{i=1}^{n} \mu_k^m \ln f_k(x_i; \theta) + \sum_{k=1}^{c} \sum_{i=1}^{n} (1-\alpha_k) \mu_k^m \ln \alpha_k
\]

subject to \( \sum_{k=1}^{c} \alpha_k = 1, \alpha_k > 0 \) and \( \sum_{k=1}^{c} \mu_k^m(x_i) = 1 \) for \( i = 1, \ldots, n \) with \( \mu_k^m(x_i) \in [0, 1] \).

The optimization problem in \( J_{\text{FCML}}(\mu, \alpha, \theta; X) \) by choosing a fuzzy \( c \)-partition \( \mu \) and a proportion \( \alpha \) and an estimate \( \theta \) to maximize \( J_{\text{FCML}}(\mu, \alpha, \theta; X) \) was called a class of FCML procedures proposed by Yang [24]. If the density \( f_k(x; \theta) \) in \( J_{\text{FCML}} \) is a multivariate standard normal distribution, Yang [24] had derived a penalized fuzzy c-means (PFCM) algorithm which is a generalized type of the well-known FCM algorithm. These PFCM had been applied in segmentation and vector quantization (see [14,15]).
We take the first derivatives of the Lagrangian $\tilde{J}_{\text{FCML}}$ with respect to the parameters $\mu_{ki}$, $\alpha_k$, $\theta_{ki}^h$ and set them to be 0. We have that

$$m\mu_{ki}^{\alpha - 1}d_{ki} - \lambda_2 = 0,$$

$$\sum_{i=1}^{n} \mu_{ki}^{\alpha}x_i - \lambda_1 \alpha_k = 0,$$

$$\sum_{i=1}^{n} \mu_{ki}^{\alpha}x_i - \lambda_3 \theta_{ki}^h = 0,$$

$$\sum_{i=1}^{c} \alpha_k = 1, \quad \sum_{i=1}^{c} \mu_{ki} = 1 \quad \text{and} \quad \sum_{h=1}^{m} \theta_{ki}^h = 1.$$ 

Based on Eqs. (18)–(21), we can derive the necessary conditions for a maximizer $(\mu, \alpha, \theta)$ of $L_{\text{FCML}}(\mu, \alpha, \theta)$ as follows:

$$\alpha_k = \frac{\sum_{i=1}^{n} \mu_{ki}^{\alpha}x_i}{\sum_{j=1}^{c} \sum_{i=1}^{n} \mu_{ki}^{\alpha}}, \quad k = 1, \ldots, c,$$

$$\theta_{ki}^h = \frac{\sum_{i=1}^{n} \mu_{ki}^{\alpha}x_i^{j^h}}{\sum_{i=1}^{n} \mu_{ki}^{\alpha}}, \quad k = 1, \ldots, c,$$

$$\mu_{ki} = \mu_k(x_i) = \left( \sum_{i=1}^{c} \left( \frac{\sum_{j=1}^{q} \sum_{h=1}^{m_j} \ln(\theta_{ki}^h)^{x_i^{j^h}} + w \ln \alpha_k}{\sum_{j=1}^{q} \sum_{h=1}^{m_j} \ln(\theta_{ki}^h)^{x_i^{j^h}} + w \ln \alpha_i} \right) \right)^{1/(m-1)}, \quad k = 1, \ldots, c, \quad i = 1, \ldots, n.$$ 

Thus the FCML clustering algorithm for categorical data is iterations through these necessary conditions of (22)–(24). In the case of $w = 0$, it becomes a fuzzy extension of the CML.

**FCML Algorithm**

**Step 1:** Fix $m \in (1, \infty)$, fix $2 \leq c \leq n$ and fix any $\alpha > 0$.

Give an initial fuzzy $c$-partition $\mu^{(0)}$ and let $s = 1$.

**Step 2:** Compute $\alpha^{(s)}$, $\theta^{(s)}$ with $\mu^{(s-1)}$ by (22) and (23).

**Step 3:** Update to $\mu^{(s)}$ with $\alpha^{(s)}$ and $\theta^{(s)}$ by (24).

**Step 4:** Compare $\mu^{(s)}$ to $\mu^{(s-1)}$ in a convenient matrix norm.

IF $\|\mu^{(s)} - \mu^{(s-1)}\| < \epsilon$, STOP.

ELSE $s = s + 1$, return to Step 2.

In Section 2, we considered three algorithms ML, EM and CML and then derived these algorithms for the estimation of parameters of multivariate multinomial mixtures with latent class models. In this section, we proposed a new approach, called FCML, using the fuzzy class model. Before we have real empirical and numerical data comparisons of these four algorithms in the next section, we should discuss more about the ML, EM, CML and FCML algorithms as follows:

(a) In these four algorithms, ML is the only one that is directly constructed on the log likelihood $L(\alpha, \theta; X)$ of a mixture of multivariate multinomial distributions with
\[ L(z, \theta; X) = \sum_{i=1}^{n} \ln \sum_{k=1}^{c} z_{ki} \prod_{j=1}^{q} \prod_{h=1}^{m_{j}} (\theta_{kh})^{x_{ij}}. \]

However, there is no closed solution to the necessary conditions for the optimization in ML approach so that we need to use the fixed-point iteration to approximate the estimates.

(b) To solve the drawback in ML method, the EM algorithm is an effective method for approximating ML estimates. Using the concept of missing data to the latent class variable \( z \), we can construct the log likelihood \( L_{EM} \) for the complete data with

\[ L_{EM}(x, \theta; x_1, \ldots, x_n, z_1, \ldots, z_n) = \sum_{i=1}^{n} \sum_{k=1}^{c} z_{ki} \{ \ln z_{ki} f_k(x_i; \theta_k) \}. \]

Thus, E-step and M-step could get the closed solution to approximate the estimates.

(c) The CML method uses a classification approach to create its log likelihood function with

\[ L_{CML}(P, x, \theta; X) = \sum_{k=1}^{c} \sum_{x_i \in P_k} \ln \{ z_{ki} f_k(x_i; \theta) \}. \]

It then uses a finite partition searching with a closed solution to approximate the estimates.

(d) The ML, EM and CML algorithms are based on mixture distributions and latent class models. However, these are considered as non-overlapping clustering and crisp class variables with a probability sense. To consider overlapping clustering with fuzzy sense, we extend latent class variables to fuzzy class variables and then create the FCML objective function \( J_{FCML} \) with

\[ J_{FCML}(\mu, x, \theta; X) = \sum_{k=1}^{c} \sum_{i=1}^{n} \mu_{ki}^{m} \ln f_k(x_i; \theta) + w \sum_{k=1}^{c} \sum_{i=1}^{n} \mu_{ki}^{m} \ln x_{ki}. \]

The closed solution for approximating the estimates can be derived based on \( J_{FCML} \).

(e) If we compare update Eqs. (11)–(13) of EM, (14) and (15) of CML and (22)–(24) of FCML, we find that the estimates of \( z_{ki} \) and \( \theta_{kh} \) have similar formula with \( z_{ki} \) as the proportion and \( \theta_{kh} \) as the weighted mean. The main difference is among these class variables \( z_{ki}, P_k \) and \( \mu_{ki} \).

(f) Because there is no closed solution in the ML algorithm, it is very difficult to consider convergence properties. However, the convergence and theoretical analysis could be investigated for the EM, CML and FCML algorithms. Similar convergence properties and analysis on EM, CML and FCML can be found from Refs. [2,21,23,24]. Recently, Yu and Yang [26] had investigated optimality test and parameter selection for the FCML method, especially for its derived algorithm PFCM.

4. Examples and numerical comparisons

In this section, we choose a real empirical dataset and some numerical data from a mixture of multivariate Bernoulli distributions. We then implement these datasets with the ML, EM, CML and FCML algorithms. On the base of three criteria with the accuracy, robustness and effectiveness, we make the comparisons of these four algorithms. The accuracy criterion is measured by the mean squared errors (MSE) which is the average sum of squared errors between the estimate and the true parameter. The robustness to initials is measured by \( \chi^2 \) values. The effectiveness of an algorithm is measured by the number of iterations (NI).

In the first example, we implement the ML, EM, CML and FCML algorithms to real data which had been analyzed by Stouffer and Toby [19] and Goodman [8]. The dataset is the observed cross-classification
of 216 respondents with respect to whether they tend toward universalistic or particularistic values in four role conflict situations. The other examples are numerical comparisons of these algorithms based on different numerical datasets according to different initials, mixing proportion $\alpha$ and parameter $\theta$ from mixture multivariate Bernoulli models. These are discussed and shown in the following examples.

4.1. Real empirical data comparisons

**Example 1.** In this example, we consider a real empirical dataset which is a 4-way contingency table with a cross-classification sample of $n$ individuals with respect to four manifest polytomous variables $A$, $B$, $C$, and $D$ presented by Stouffer and Toby [19]. If there is, some latent dichotomous variable $X$, so that each of the $n$ individuals is in one of the two latent classes with respect to this variable, and within the $c$th latent class the manifest variables $(A, B, C, D)$ are mutually independent, then this two-class latent structure would serve as a simple explanation of the observed relationships among the variables in the 4-way contingency table for the $n$ individuals. There is a direct generalization when the latent variable has $c$ classes. We shall present some relatively simple methods for determining whether the observed relationships among the variables in the $m$-way contingency table can be explained using a $c$-class structure, or by various modifications and extensions of this latent structure.

To illustrate these methods we analyze Table 1, a $2^4$ contingency table presented earlier by Stouffer and Toby [19] and analyzed by Goodman [8], which cross-classifies 216 respondents with respect to their tendency towards universalistic values (1) or particularistic values (0) when confronted by each of four different role conflict situations. The letters $A$, $B$, $C$, and $D$ in Table 1 denote the dichotomous responses when confronted by the four different situations. We perform these different algorithms by trying various initial values of $a_k$. We compare the estimations obtained by the algorithms using the chi-squared statistic $\chi^2$ based upon the likelihood ratio which can measure the difference between the empirical data and the estimation

$$\chi^2 = 2 \sum_{i=0}^{15} f_i \log \left( \frac{f_i}{nF_i} \right).$$

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>Observed frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>42</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>23</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>25</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>24</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>7</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>38</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>9</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
</tr>
</tbody>
</table>
where

\[ \hat{F}_i = \sum_{k=1}^{2} \alpha_k \prod_{j=1}^{4} (\theta_{ik}^j)^{y_{ij}} (1 - \theta_{ik}^j)^{1 - y_{ij}} \]

with \( f_i \) being the observed frequency (see Goodman [8]).

We give one group of initial values for \( \theta_{ik}^j \) as in Table 2 and five groups of different initial values for \( \alpha_k \) in Table 3. We implement the MLE, EM, CML and FCML algorithms to the data set in Table 1 under the initial values in Tables 2 and 3. On the base of \( \chi^2 \) values and NI values, the robustness and effectiveness of these algorithms for estimating the parameters of the multivariate binomial mixtures are described as follows:

(a) **ML algorithm:** In five groups of initial values, we obtain various results and quite different \( \chi^2 \) values using the ML algorithm, as shown in Table 4. However, the ML algorithm can get a very low \( \chi^2 \) value of 2.9. The results tell us that the MLE algorithm can get very good estimates under good initial values, but it is very sensitive to initial values. The robustness of the ML algorithm is not good enough. The number of iterations (NI) can be low but not stable.

(b) **EM algorithm:** The results obtained using the EM algorithm are shown in Table 5. The \( \chi^2 \) values change a lot depending upon the initial values. This algorithm is sensitive to the initial values but can get a good estimation with good initial values. The NI values are very high. That is, the EM algorithm is a high time-consumer.

(c) **CML algorithm:** The results are shown in Table 6. The \( \chi^2 \) values of the CML algorithm are high. That is, the estimation is not good enough. However, the \( \chi^2 \) values are stable. The CML algorithm is also robust. The NI values are all low.

(d) **FCML algorithm:** The results are shown in Table 7. The estimation of the FCML (with \( w = 0 \)) algorithm is not as good as that of the ML and EM. The \( \chi^2 \) values of the FCML algorithm are all the same,

### Table 2
**Different initial values of \( \theta \)**

<table>
<thead>
<tr>
<th>( k )</th>
<th>( \theta_{1k}^1 )</th>
<th>( \theta_{1k}^2 )</th>
<th>( \theta_{1k}^3 )</th>
<th>( \theta_{1k}^4 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.7</td>
<td>0.5</td>
<td>0.8</td>
<td>0.7</td>
</tr>
<tr>
<td>2</td>
<td>0.9</td>
<td>0.3</td>
<td>0.4</td>
<td>0.4</td>
</tr>
</tbody>
</table>

### Table 3
**Different initial values of \( \alpha \)**

<table>
<thead>
<tr>
<th>( k )</th>
<th>( \alpha_k )</th>
</tr>
</thead>
<tbody>
<tr>
<td>I 1</td>
<td>0.3</td>
</tr>
<tr>
<td>2</td>
<td>0.7</td>
</tr>
<tr>
<td>II 1</td>
<td>0.4</td>
</tr>
<tr>
<td>2</td>
<td>0.6</td>
</tr>
<tr>
<td>III 1</td>
<td>0.1</td>
</tr>
<tr>
<td>2</td>
<td>0.9</td>
</tr>
<tr>
<td>IV 1</td>
<td>0.7</td>
</tr>
<tr>
<td>2</td>
<td>0.3</td>
</tr>
<tr>
<td>V 1</td>
<td>0.9</td>
</tr>
<tr>
<td>2</td>
<td>0.1</td>
</tr>
</tbody>
</table>
so it is very robust. However, it is actually an improved CML algorithm. We know that the CML algorithm is derived under latent class models but the FCML algorithm is derived under fuzzy class models. Let us

Table 4
Results of five initial values using the ML algorithm

<table>
<thead>
<tr>
<th>Group</th>
<th>$k$</th>
<th>$x_1^k$</th>
<th>$\theta_1^k$</th>
<th>$\theta_2^k$</th>
<th>$\theta_3^k$</th>
<th>$\theta_4^k$</th>
<th>$\chi^2$</th>
<th>NI</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>1</td>
<td>0.298</td>
<td>0.982</td>
<td>0.923</td>
<td>0.912</td>
<td>0.748</td>
<td>2.9</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0.701</td>
<td>0.706</td>
<td>0.318</td>
<td>0.344</td>
<td>0.123</td>
<td></td>
<td></td>
</tr>
<tr>
<td>II</td>
<td>1</td>
<td>0.292</td>
<td>0.932</td>
<td>0.925</td>
<td>0.912</td>
<td>0.748</td>
<td>4.3</td>
<td>26</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0.708</td>
<td>0.706</td>
<td>0.368</td>
<td>0.354</td>
<td>0.173</td>
<td></td>
<td></td>
</tr>
<tr>
<td>III</td>
<td>1</td>
<td>0.001</td>
<td>0.982</td>
<td>0.922</td>
<td>0.912</td>
<td>0.798</td>
<td>160.7</td>
<td>156</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0.999</td>
<td>0.906</td>
<td>0.118</td>
<td>0.744</td>
<td>0.183</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IV</td>
<td>1</td>
<td>0.545</td>
<td>0.762</td>
<td>0.653</td>
<td>0.352</td>
<td>0.658</td>
<td>88.7</td>
<td>57</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0.455</td>
<td>0.886</td>
<td>0.438</td>
<td>0.664</td>
<td>0.193</td>
<td></td>
<td></td>
</tr>
<tr>
<td>V</td>
<td>1</td>
<td>1</td>
<td>0.842</td>
<td>0.009</td>
<td>0.234</td>
<td>0.848</td>
<td>199.1</td>
<td>324</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0</td>
<td>0.067</td>
<td>0.318</td>
<td>0.934</td>
<td>0.145</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5
Results of five initial values using the EM algorithm

<table>
<thead>
<tr>
<th>Group</th>
<th>$k$</th>
<th>$x_1^k$</th>
<th>$\theta_1^k$</th>
<th>$\theta_2^k$</th>
<th>$\theta_3^k$</th>
<th>$\theta_4^k$</th>
<th>$\chi^2$</th>
<th>NI</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>1</td>
<td>0.391</td>
<td>0.960</td>
<td>0.920</td>
<td>0.901</td>
<td>0.721</td>
<td>9.25</td>
<td>1031</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0.609</td>
<td>0.706</td>
<td>0.321</td>
<td>0.287</td>
<td>0.093</td>
<td></td>
<td></td>
</tr>
<tr>
<td>II</td>
<td>1</td>
<td>0.454</td>
<td>0.763</td>
<td>0.652</td>
<td>0.818</td>
<td>0.636</td>
<td>18.8</td>
<td>1648</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0.546</td>
<td>0.834</td>
<td>0.703</td>
<td>0.424</td>
<td>0.196</td>
<td></td>
<td></td>
</tr>
<tr>
<td>III</td>
<td>1</td>
<td>0.009</td>
<td>0.663</td>
<td>0.542</td>
<td>0.527</td>
<td>0.616</td>
<td>196.5</td>
<td>4032</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0.991</td>
<td>0.893</td>
<td>0.231</td>
<td>0.121</td>
<td>0.100</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IV</td>
<td>1</td>
<td>0.172</td>
<td>0.712</td>
<td>0.824</td>
<td>0.632</td>
<td>0.631</td>
<td>107.8</td>
<td>2071</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0.828</td>
<td>0.623</td>
<td>0.432</td>
<td>0.134</td>
<td>0.111</td>
<td></td>
<td></td>
</tr>
<tr>
<td>V</td>
<td>1</td>
<td>0.999</td>
<td>0.673</td>
<td>0.592</td>
<td>0.548</td>
<td>0.521</td>
<td>263.5</td>
<td>4631</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0.001</td>
<td>0.801</td>
<td>0.227</td>
<td>0.119</td>
<td>0.001</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 6
Results of five initial values using the CML algorithm

<table>
<thead>
<tr>
<th>Group</th>
<th>$k$</th>
<th>$x_1^k$</th>
<th>$\theta_1^k$</th>
<th>$\theta_2^k$</th>
<th>$\theta_3^k$</th>
<th>$\theta_4^k$</th>
<th>$\chi^2$</th>
<th>NI</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>1</td>
<td>0.502</td>
<td>0.792</td>
<td>0.504</td>
<td>0.534</td>
<td>0.366</td>
<td>82.05</td>
<td>23</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0.497</td>
<td>0.801</td>
<td>0.475</td>
<td>0.504</td>
<td>0.287</td>
<td></td>
<td></td>
</tr>
<tr>
<td>II</td>
<td>1</td>
<td>0.494</td>
<td>0.824</td>
<td>0.474</td>
<td>0.525</td>
<td>0.226</td>
<td>88.8</td>
<td>41</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0.505</td>
<td>0.808</td>
<td>0.484</td>
<td>0.515</td>
<td>0.242</td>
<td></td>
<td></td>
</tr>
<tr>
<td>III</td>
<td>1</td>
<td>0.497</td>
<td>0.768</td>
<td>0.484</td>
<td>0.484</td>
<td>0.315</td>
<td>82.51</td>
<td>22</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0.502</td>
<td>0.760</td>
<td>0.5</td>
<td>0.5</td>
<td>0.323</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IV</td>
<td>1</td>
<td>0.5</td>
<td>0.836</td>
<td>0.491</td>
<td>0.465</td>
<td>0.336</td>
<td>84.83</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0.5</td>
<td>0.827</td>
<td>0.508</td>
<td>0.491</td>
<td>0.310</td>
<td></td>
<td></td>
</tr>
<tr>
<td>V</td>
<td>1</td>
<td>0.5</td>
<td>0.756</td>
<td>0.432</td>
<td>0.532</td>
<td>0.243</td>
<td>91.28</td>
<td>46</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0.5</td>
<td>0.774</td>
<td>0.432</td>
<td>0.540</td>
<td>0.243</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
consider the FCML with $w = 0.3$. The results are shown in Table 8. The $\chi^2$ values of the PFCML (with $w = 0.3$) algorithm are very stable and its estimation is better and also improves the CML and FCML (with $w = 0$) algorithms. We see that the penalty term with the weight $w$ in the FCML can be an adjustment of the bias of the FCML algorithm.

In this real data comparison of the ML, EM, CML and FCML algorithms, we find that the CML and FCML algorithms are robust to initial values. However, the FCML presents better results. The MLE and EM give very good results if we give good initial values. Overall, the proposed FCML algorithm presents better robustness and effectiveness than MLE, EM and CML algorithms based on the $\chi^2$ values and NI values for this real empirical data set.

### 4.2. Numerical data comparisons

For the following Examples 2–4, we use these algorithms to estimate the parameters of multivariate Bernoulli mixtures using random samples drawn from the mixtures distribution $f(x; \theta)$ given by
Example 2. We consider a random sample from the mixture distribution \( f(x; \theta) \) given by

\[
f(x; \theta) = 0.4B(1, 0.9)B(1, 0.7)B(1, 0.4)B(1, 0.3) + 0.6B(1, 0.4)B(1, 0.3)B(1, 0.5)B(1, 0.8),
\]

where \( B(1, p) \) is a Bernoulli distribution. The random sample is shown in Table 9. We assign nine different initial values for \( a \) as (I) \( a = 0.1 \), (II) \( a = 0.2 \), (III) \( a = 0.3 \), (IV) \( a = 0.4 \), (V) \( a = 0.5 \), (VI) \( a = 0.6 \), (VII) \( a = 0.7 \), (VIII) \( a = 0.8 \), (IX) \( a = 0.9 \). Table 10 shows the mean squared errors (MSE) between the estimate and true parameter from the EM, CML and FCML algorithms under these different initial values.

We find that the EM algorithm gets the lowest MSE for two initial values of \( a \). The CML algorithm get one lowest MSE for an initial value of \( a \). Although the FCML algorithm without the penalty term (i.e. \( w = 0 \)) does not get the lowest MSE, it is quite stable (or robust) to the different initial values of \( a \). Overall, the FCML algorithm with \( w = 0.3 \) gets most of the lowest MSE and has the lowest MSE 0.033 in average.

Example 3. We consider the mixture model

\[
f(x; \theta) = \alpha B(1, 0.9)B(1, 0.7)B(1, 0.4)B(1, 0.3) + (1 - \alpha)B(1, 0.4)B(1, 0.3)B(1, 0.5)B(1, 0.8)
\]

with different proportion \( \alpha \) of (A1) \( \alpha = 0.1 \), (A2) \( \alpha = 0.4 \), (A3) \( \alpha = 0.7 \). For tests A1, A2 and A3, we implement the algorithms for the drawn random samples under nine different initial values as in Example 2. The MSE and NI values from the different algorithms are shown in Table 11.

In this numerical example, we consider two criteria of accuracy and effectiveness with MSE and NI values. We find that the FCML with \( w = 0 \) has most effectiveness with lowest NI values. However, the FCML with \( w = 0.3 \) has most accuracy with the lowest MSE. We known that the difference between FCML with \( w = 0 \) and \( w = 0.3 \) is the penalty term.

Example 4. We consider mixture distribution \( f(x; \theta) \) given by

\[
f(x; \theta) = \sum_{k=1}^{2} \sum_{h=1}^{4} (\theta_k^h)^x (1 - \theta_k^h)^{1-x}.
\]
with different parameter $h_k$ for $k = 1, 2, h = 1, 2, 3, 4$, and with tests B1 and B2 shown in Table 12. For tests B1 and B2, we implement these algorithms for the drawn random samples under nine different initial values as in Example 1. We give their MSE and NI values which are shown in Table 13. For these two sets of different parameters $\theta$, we get very impressive MSE with 0.004 and 0.003 when we apply the FCML algorithm with $w = 0.3$ to the estimation of parameters in the mixture multivariate Bernoulli distributions.

Table 10
MSE from different algorithms

<table>
<thead>
<tr>
<th>Test</th>
<th>EM</th>
<th>CML</th>
<th>FCML (with $w = 0$)</th>
<th>FCML (with $w = 0.3$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>0.036</td>
<td>0.036</td>
<td>0.037</td>
<td><strong>0.030</strong></td>
</tr>
<tr>
<td>II</td>
<td><strong>0.033</strong></td>
<td>0.037</td>
<td>0.037</td>
<td>0.036</td>
</tr>
<tr>
<td>III</td>
<td><strong>0.035</strong></td>
<td>0.049</td>
<td>0.037</td>
<td>0.036</td>
</tr>
<tr>
<td>IV</td>
<td>0.042</td>
<td>0.039</td>
<td>0.037</td>
<td><strong>0.035</strong></td>
</tr>
<tr>
<td>V</td>
<td>0.050</td>
<td>0.039</td>
<td>0.037</td>
<td><strong>0.035</strong></td>
</tr>
<tr>
<td>VI</td>
<td>0.041</td>
<td>0.049</td>
<td>0.037</td>
<td><strong>0.026</strong></td>
</tr>
<tr>
<td>VII</td>
<td>0.051</td>
<td><strong>0.029</strong></td>
<td>0.037</td>
<td>0.034</td>
</tr>
<tr>
<td>VIII</td>
<td>0.064</td>
<td>0.039</td>
<td>0.037</td>
<td><strong>0.035</strong></td>
</tr>
<tr>
<td>IX</td>
<td>0.080</td>
<td>0.040</td>
<td>0.037</td>
<td><strong>0.036</strong></td>
</tr>
<tr>
<td>Average</td>
<td>0.048</td>
<td>0.036</td>
<td>0.037</td>
<td><strong>0.033</strong></td>
</tr>
</tbody>
</table>

Table 11
MSE and NI from different algorithms

<table>
<thead>
<tr>
<th>Test</th>
<th>EM</th>
<th>CML</th>
<th>FCML (with $w = 0$)</th>
<th>FCML (with $w = 0.3$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSE</td>
<td>NI</td>
<td>MSE</td>
<td>NI</td>
<td>MSE</td>
</tr>
<tr>
<td>A1</td>
<td>0.048</td>
<td>1029</td>
<td>0.036</td>
<td>473</td>
</tr>
<tr>
<td>A2</td>
<td>0.097</td>
<td>1487</td>
<td>0.084</td>
<td>436</td>
</tr>
<tr>
<td>A3</td>
<td>0.065</td>
<td>1211</td>
<td>0.054</td>
<td>432</td>
</tr>
</tbody>
</table>

$f(x; \theta) = 0.5B(1, \theta_1^1)B(1, \theta_1^2)B(1, \theta_1^3)B(1, \theta_1^4) + 0.5B(1, \theta_2^1)B(1, \theta_2^2)B(1, \theta_2^3)B(1, \theta_2^4)$

Table 12
For fixed $x = 0.5$, two groups of initial values of $\theta$

<table>
<thead>
<tr>
<th>Test</th>
<th>Class $k$</th>
<th>$\theta_1^1$</th>
<th>$\theta_1^2$</th>
<th>$\theta_1^3$</th>
<th>$\theta_1^4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>B1</td>
<td>1</td>
<td>0.9</td>
<td>0.7</td>
<td>0.4</td>
<td>0.3</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0.4</td>
<td>0.3</td>
<td>0.5</td>
<td>0.8</td>
</tr>
<tr>
<td>B2</td>
<td>1</td>
<td>0.5</td>
<td>0.6</td>
<td>0.9</td>
<td>0.2</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0.7</td>
<td>0.2</td>
<td>0.4</td>
<td>0.8</td>
</tr>
</tbody>
</table>

Table 13
MSE and NI from different algorithms

<table>
<thead>
<tr>
<th>Test</th>
<th>EM</th>
<th>CML</th>
<th>FCML (with $w = 0$)</th>
<th>FCML (with $w = 0.3$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSE</td>
<td>NI</td>
<td>MSE</td>
<td>NI</td>
<td>MSE</td>
</tr>
<tr>
<td>B1</td>
<td>0.057</td>
<td>127</td>
<td>0.033</td>
<td>78</td>
</tr>
<tr>
<td>B2</td>
<td>0.065</td>
<td>213</td>
<td>0.040</td>
<td>79</td>
</tr>
</tbody>
</table>
5. Conclusions

In this paper, we presented clustering algorithms for analyzing grouped categorical data. We extended a latent class model into a fuzzy class model and then created its corresponding algorithm called the FCML. We know that the latent class variables are indicator functions with the crisp values \{0, 1\}. To enhance class variables more robust, we used fuzzy class variables and then embedded the fuzzy clustering algorithm with a penalized term for mixture multivariate multinomial distribution models. We conclude that this fuzzy algorithm FCML could be successfully used in estimating the parameters of multivariate binomial mixtures. We provided the comparisons under accuracy criteria (i.e. MSE), robustness (i.e. $\chi^2$ values) and computational efficiency (i.e. NI) for the ML, EM, CML, FCML algorithms. According to the numerical comparisons, we understand that the different algorithms have accuracy and computational efficiency. Of course, each algorithm has its special strength. However, the proposed fuzzy algorithm FCML seems to have better accuracy, robustness and effectiveness. The superiority of FCML over ML, EM and CML is shown using a real empirical dataset and different random sample from different types of multivariate Bernoulli mixtures. We recommend those concerned with applications in latent class models for categorical data to try to use this proposed fuzzy algorithm with a fuzzy class model for categorical data. In the marketing management, market segmentation has became a central concern of top management and strategic planners (see Wedel and Kamakura [20]). Cluster analysis has been considered in market segmentation. The applications of the proposed algorithms to market segmentation and also logistic regression will be our further research.

Acknowledgements

The authors are grateful to the anonymous referees for their helpful comments and suggestions to improve the presentation of the paper. This work was supported in part by the National Science Council of Taiwan, ROC, under grant NSC-88-2118-M-033-001.

References

[9] B.F. Green, A general solution for the latent class model of latent structure analysis, Psychometrika 16 (1951) 151–166.


