International Journal of Computational Intelligence and Applications
Vol. 14, No. 1 (2015) 1550001 (16 pages)

DOI: 10.1142/S1469026815500017

Imperial College Press
www.icpress.co.uk

Collaborative Fuzzy Clustering of Variational Bayesian Generative Topographic Mapping

Mohamad Ghassany* and Younès Bennani†

LIPN, CNRS 7030
Université Paris 13, Sorbonne Paris Cité
Villetaneuse, France
**m.ghassany@gmail.com*
†*younes.bennani@lipn.univ-paris13.fr*

Received 2 September 2013
Revised 29 October 2014
Published 30 March 2015

In this paper, we propose a Collaborative Clustering method based on Variational Bayesian Generative Topographic Mapping (VBGTM). To do so, we first propose a method that combines VBGTM and Fuzzy c-means (FCM). Collaborative clustering is useful to achieve interaction between different sources of information for the purpose of revealing underlying structures and regularities within data sets. It can be treated as a process of consensus building where we attempt to reveal a structure that is common across all sets of data. VBGTM was introduced as a variational approximation of Generative Topographic Mapping (GTM) to control data overfitting. It provides an analytical approximation to the posterior probability of the latent variables and the distribution of the input data in the latent space. It can be effectively applied to visualize and explore properties of the data. But when the number of latent points is large, similar units need to be grouped (i.e., clustered) to facilitate quantitative analysis of the map and the data. We use FCM to determine the prototypes as well as the resultant clusters and the corresponding membership functions of the input data, based on the latent variables obtained from VBGTM. So, by combining the two algorithms, we develop a method that can do visualization and clustering at the same time. We observe that the hybrid method (F-VBGTM) performs very well in terms of many cluster-validity indexes.

Keywords: Collaborative clustering; fuzzy clustering; generative topographic mapping; variational inference.

1. Introduction

Cluster analysis divides data into groups (clusters) such that similar data objects belong to the same cluster and dissimilar data objects to different clusters. The clustering problem has been addressed in many contexts and by researchers in many disciplines; this reflects its broad appeal and usefulness as one of the steps in exploratory data analysis. Fuzzy clustering algorithms are important members of the family of clustering algorithms. Compared to hard clustering or crisp clustering, fuzzy clustering can reveal the differences between data vectors more explicitly. This can be noticed due to the role of fuzzy clustering in assigning vectors to clusters and

determines a membership function that indicates how likely the data vector belongs to a cluster. The most prominent fuzzy clustering algorithm is the fuzzy c-means (FCM),[5] a fuzzification of k-means.[16] But FCM suffers from the curse of dimensionality, it cannot visualize data if the dimension is bigger than three.

Besides the FCM algorithm, other techniques perform visualization and clustering with membership, such as Generative Topographic Mapping (GTM).[6,7] GTM was proposed as a probabilistic counterpart to the Self-Organizing Maps (SOM).[15] GTM is mostly used for data visualization since it allows high dimensional data to be modeled as resulting from Gaussian noise added to sources in lower-dimensional latent space. But in its basic formulation, the GTM is trained within the Maximum Likelihood (ML) framework using the Expectation-Maximization (EM) algorithm, permitting data overfitting unless regularization is included. A Variational Bayesian approach of the GTM (VBGTM) was introduced in Ref. 18 to endow the model with regularization capabilities based on variational techniques. But when the number of latent points is large, similar units need to be clustered to facilitate quantitative analysis of the map and the data. That's why we propose to extend the VBGTM to a fuzzy clustering technique so that visualization of clustering results is possible together with probabilistic clustering.

So far, clustering techniques described above operate on a single data set. Nowadays, computing environments and technologies are more and more evolving towards a mobile, finely distributed, interacting, dynamic environment containing massive amounts of heterogeneous, spatially and temporally distributed data sources. In many companies data is distributed among several sites, i.e., each site generates its own data and manages its own data repository. Analyzing these distributed sources requires distributed clustering techniques to find global patterns representing the complete information. The transmission of the entire local data set is often unacceptable because of performance considerations, privacy and security aspects, and bandwidth constraints. Traditional clustering algorithms, demanding access to complete data, are not appropriate for distributed applications. Thus, there is a need for distributed clustering algorithms in order to analyze and discover new knowledge in distributed environments.

In this paper, we propose a new method for fuzzy clustering that combines VBGTM and FCM and takes advantages of the features of these two techniques, we call it F-VBGTM, then extend it to be applied on distributed data sets. In Sec. 2, we discuss the related literature, more precisely the FCM, GTM and VBGTM methods. We move on to the description of the proposed method and its advantages in Sec. 3. We compare the performance of the method with the standard FCM in Sec. 4. We apply our method to Collaborative clustering in Sec. 5 and we terminate with a conclusion.

2. Literature Review

2.1. *Fuzzy clustering*

The FCM algorithm[5] is one of the most widely used methods in fuzzy clustering. It is based on the concept of fuzzy c-partition,[20] summarized as follows.

Let $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ be a set of given data, where each data point $\mathbf{x}_n (n = 1, \ldots, N)$ is a vector in $\mathbb{R}^D$, U is a $C \times N$ matrix, and C be an integer, $2 \leq C < N$. Then, the fuzzy C-partition space for X is the set

$$M_{f_{cn}} = \left\{ U \in \mathbb{R}^{C \times N} : \mu_{cn} \in [0,1], \sum_{c=1}^{C} \mu_{cn} = 1, 0 < \sum_{n=1}^{N} \mu_{cn} < N \right\}, \tag{1}$$

where μ_{cn} is the membership value of $\mathbf{x}_n$ in cluster $c (c = 1, \ldots, C)$. The aim of the FCM algorithm is to find an optimal fuzzy c-partition and corresponding prototypes minimizing the objective function

$$J_m(U; V; X) = \sum_{n=1}^{N} \sum_{c=1}^{C} (\mu_{cn})^m \|\mathbf{x}_n - \nu_c\|^2. \tag{2}$$

In Eq. (2), $V = (\nu_1, \ldots, \nu_C)$ is a matrix of unknown cluster centres (prototypes; seeds) $\nu_c \in \mathbb{R}^D$, $\|\cdot\|$ is the Euclidean norm, and m is a fuzzifier in $[1, \infty]$, it influences the membership values and is usually chosen to be two.

The cluster centres and the respective membership functions that solve the constrained optimization problem in (2) are given by the following equations:

$$\nu_c = \frac{\sum_{n=1}^{N} (\mu_{cn})^m \mathbf{x}_n}{\sum_{n=1}^{N} (\mu_{cn})^m}, \quad 1 \leq c \leq C, \tag{3}$$

$$\mu_{cn} = \left[\sum_{j=1}^{C} \left(\frac{\|\mathbf{x}_n - \nu_c\|^2}{\|\mathbf{x}_n - \nu_j\|^2} \right)^{1/(m-1)} \right]^{-1}, \quad 1 \leq c \leq C, \quad 1 \leq n \leq N. \tag{4}$$

Equations (3) and (4) constitute an iterative optimization procedure. The goal is to iteratively improve a sequence of sets of fuzzy clusters until no further improvement in J_m is possible.

An overview and comparison of different fuzzy clustering algorithms is available.[17]

2.2. *The GTM standard model*

GTM is defined as a mapping from a low dimensional latent space $\mathbb{R}^L$ (with L being usually 1 or 2 for visualization purposes) onto the observed data space $\mathbb{R}^D$. The mapping is carried through by a set of basis functions generating a constrained mixture density distribution. It is defined as a generalized linear regression model:

$$y = y(z, W) = W\Phi(z), \tag{5}$$

where $y \in \mathbb{R}^D$, $z \in \mathbb{R}^L$ and Φ is a matrix consisting of M basis functions $(\phi_1(z), \ldots, \phi_M(z))$, introducing the non-linearity, W is a $D \times M$ matrix of adaptive weights w_{dm} that defines the mapping. The standard definition of GTM considers spherically symmetric Gaussians as basis functions. To achieve computational tractability, the prior distribution of z in latent space is constrained to form a uniform discrete grid of K centres, analogous to the layout of the SOM[15] units, in

the form:

$$p(z) = \frac{1}{K} \sum_{i=1}^{K} \delta(z - z_i). \tag{6}$$

GTM can also be understood as a constrained mixture of Gaussians. The corresponding y_k of each latent variable z_k represents the centre of a Gaussian density function, which is the mean of the Gaussian distribution. Assuming that the observed data set X consists of N i.i.d. data points x_n, this leads to the definition of a complete likelihood of the form:

$$\begin{aligned} \mathcal{L}(W, \beta) &= \ln \prod_{n=1}^{N} p(x_n|W, \beta) \\ &= \sum_{n=1}^{N} \ln \left\{ \frac{1}{K} \sum_{i=1}^{K} p(x_n|z_i, W, \beta) \right\}, \end{aligned} \tag{7}$$

where $y_k = W\Phi(z_k)$ are the reference vectors. The parameters W and β can be optimized by ML using the EM algorithm.

By calculating the mean or mode of the posterior probabilities of the latent variables, the distribution of input data can be visualized in the latent space.

This version of GTM uses the ML method to estimate its model parameters. However, as Svensèn remarked in his PhD thesis,[21] it is too susceptible to overfit the data. A regularized version of the GTM using the evidence approximation was in fact introduced in that work. A Markov Chain Monte Carlo (MCMC) method using Gibbs sampling,[22] as well as a first approximation using a variational framework, were applied to improve the parameter estimation of the GTM model in Ref. 23. In Ref. 18, a full variational version for the GTM was presented based on the GTM with a Gaussian process (GP) prior outlined in Ref. 7, to which a Bayesian estimation of the parameters is added. We now show the importance of using the variational approach for GTM.

2.3. *Variational inference*

The central idea of variational Bayesian inference[8] is to introduce a set of distributions over the parameters into the marginal likelihood, in such a way that the computation of the marginal likelihood $p(\mathbf{X}) = \int p(\mathbf{X}|\Theta)p(\Theta)d\Theta$ becomes tractable, $\Theta = \{\theta_i\}$ is the set of parameters defining the model. Variational Bayesian inference has quickly become a popular way to learn otherwise intractable models (See Refs. 1, 10, 14). In the context of Bayesian inference, this framework is known as *variational Bayes*.

The starting point of the variational Bayesian framework is the marginal likelihood, which, in logarithmic form, can be expressed as follows:

$$\ln p(\mathbf{X}) = \ln \frac{p(\mathbf{X}, \Theta)}{p(\Theta|\mathbf{X})}, \tag{8}$$

where the model structure $\mathcal{M}$ is assumed to be implicit. At this point, a distribution q over the parameters Θ can be introduced, which will be henceforth called *variational distribution*, given that the log marginal likelihood does not depend on Θ:

$$\ln p(\mathbf{X}) = \int q(\Theta) \ln \frac{p(\mathbf{X}, \Theta)}{p(\Theta|\mathbf{X})} d\Theta. \tag{9}$$

After some mathematical transformations, Eq. (9) can be rewritten as:

$$\begin{aligned} \ln p(\mathbf{X}) = & \int q(\Theta) \ln \frac{p(\mathbf{X}, \Theta)}{q(\Theta)} d\Theta + \int q(\Theta) \ln \frac{q(\Theta)}{p(\Theta|\mathbf{X})} d\Theta \\ & = F(q(\Theta)) + D_{KL}[q(\Theta)||p(\Theta|\mathbf{X})], \end{aligned} \tag{10}$$

where $D_{KL}[q(\Theta)||p(\Theta|\mathbf{X})]$ is the Kullback–Leibler (KL) divergence between the variational and the posterior distributions. Given that KL divergence is a strictly non-negative term, $F(q(\Theta))$ becomes a lower bound function on the log marginal likelihood. As a result, the convergence of the former guarantees the convergence of the latter:

$$\ln p(\mathbf{X}) \geq F(q(\Theta)). \tag{11}$$

Thus, the ultimate goal in variational Bayesian inference is choosing a suitable form for the variational distribution $q(\Theta)$ in such a way that $F(q)$ can be readily evaluated and yet which is sufficiently flexible that the bound is reasonably tight.

In the case of latent variable models, the latent or hidden variables $\mathbf{Z}$ can be easily incorporated into the variational Bayesian framework as an additional set of model parameters. In this manner, a prior distribution $p(\mathbf{Z})$ over the hidden variables will be also required.

Taking as inspiration the EM algorithm,[9] an efficient variational Bayesian expectation-maximization (VBEM) algorithm[4] that could be applied to many Statistical Machine Learning (SML) latent variable models can be defined by assuming independent variational distributions over $\mathbf{Z}$ and Θ, i.e., $q(\mathbf{Z}, \Theta) = q(\mathbf{Z})q(\Theta)$. Thereby, the VBEM algorithm can be derived by maximization of F as follows:
VBE-Step:

$$q(\mathbf{Z})^{(\text{new})} \leftarrow \underset{q(\mathbf{Z})}{\text{argmax}}\, F(q(\mathbf{Z})^{(\text{old})}, q(\Theta)), \tag{12}$$

VBM-Step:

$$q(\Theta)^{(\text{new})} \leftarrow \underset{q(\Theta)}{\text{argmax}}\, F(q(\mathbf{Z})^{(\text{new})}, q(\Theta)). \tag{13}$$

In summary, variational Bayesian inference offers an elegant framework in which approximate inference can be performed in a closed way, and which allows efficient Bayesian inference of the model parameters and hidden variables. Next section applies these concepts to GTM to yield new powerful analytic methods.

2.4. *Variational Bayesian GTM*

2.4.1. *Bayesian formulation of GTM*

The specification of a full Bayesian model of GTM can be obtained by defining priors over the parameters $\mathbf{Y}, \mathbf{Z}$ and β. In Ref. 7, a prior for $\mathbf{Y}$ was proposed, the regression function using basis functions is replaced by a smooth mapping carried out by a GP prior, since the original GTM has a hard constraint imposed on the mapping from latent space to the data space due to the finite number of basis functions used. This way the likelihood takes the form,

$$p(\mathbf{X}|\mathbf{Z}, \mathbf{Y}, \beta) = \left(\frac{\beta}{2\pi}\right)^{ND/2} \prod_{n=1}^{N} \prod_{k=1}^{K} \left\{\exp\left(-\frac{\beta}{2}\|\mathbf{x}_n - \mathbf{y}_k\|^2\right)\right\}^{z_{nk}}, \tag{14}$$

where $\mathbf{Z} = \{z_{kn}\}$ are binary membership variables complying with the restriction $\sum_{k=1}^{K} z_{kn} = 1$, and y_k are the column vectors of a matrix $\mathbf{Y}$ and the centroids of spherical Gaussian generators equivalent to the reference vectors of the GTM. The prior defined over $\mathbf{Y}$ is a GP prior defined as:

$$p(\mathbf{Y}) = (2\pi)^{-KD/2}|\mathbf{C}|^{-D/2} \prod_{d=1}^{D} \exp\left(-\frac{1}{2}\mathbf{y}_{(d)}^{T}\mathbf{C}^{-1}\mathbf{y}_{(d)}\right), \tag{15}$$

where $\mathbf{y}_{(d)}$ is each of the row vectors of the matrix $\mathbf{Y}$, and $\mathbf{C}$ is a matrix where each of its elements is a covariance function defined as:

$$\mathbf{C}(i,j) = \mathbf{C}(z_i, z_j) = \epsilon \exp\left(-\frac{\|z_i - z_j\|^2}{2\alpha^2}\right), \quad i,j = 1,\ldots,K, \tag{16}$$

and where hyperparameter ϵ is usually set to a value of 1. The α hyperparameter controls the flexibility of the mapping from the latent space to the data space. A suitable choice for prior distributions will yield a tractable variational Bayesian solution. Since z_{kn} are defined as binary values, a multinomial distribution can be chosen for $\mathbf{Z}$:

$$p(\mathbf{Z}) = \prod_{n=1}^{N} \prod_{k=1}^{K} \gamma_{kn}^{z_{kn}}, \tag{17}$$

where γ_{kn} is an hyperparameter controlling the distribution over each of z_{kn}. Finally, a Gamma distribution is chosen to be the prior over β:

$$p(\beta) = \Gamma(\beta|d_\beta, s_\beta), \tag{18}$$

where d_β and s_β are the hyperparameters of the parameter β. A graphical representation of the Bayesian GTM, including the hidden variables, parameters and hyperparameters, is shown in Fig. 1.

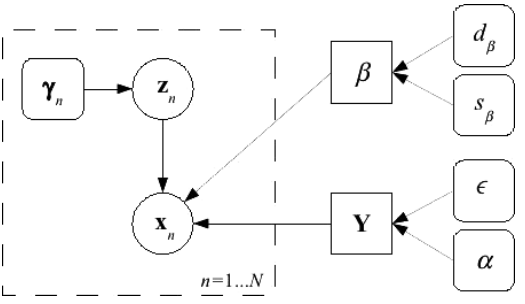


Fig. 1. Graphical model representation of the Bayesian GTM.

2.4.2. *Variational Bayesian approach for GTM (VBGTM)*

As described above, variational inference allows approximating the marginal log-likelihood through Jensen's inequality:

$$\ln p(\mathbf{X}) \geq F(q(\mathbf{Z}, \Theta)). \tag{19}$$

The function $F(q(\mathbf{Z}, \Theta))$ is a lower bound such that its convergence guarantees the convergence of the marginal likelihood. The goal is choosing a suitable form for the variational distribution $F(q(\mathbf{Z}, \Theta))$ in such way that $F(q)$ can be readily evaluated. We assume that the hidden membership variable $\mathbf{Z}$ and the parameters Θ are i.i.d., i.e., $q(\mathbf{Z}, \Theta) = q(\mathbf{Z})q(\Theta)$. Thereby, a Variational EM algorithm can be derived[4]: a *VBE step* as mentioned in Eq. (12) and a *VBM step* as in Eq. (13).

2.4.3. *VBE step*

The form chosen for the variational distribution $q(\mathbf{Z})$ is similar to that of the prior distribution $p(\mathbf{Z})$:

$$p(\mathbf{Z}) = \prod_{n=1}^{N} \prod_{k=1}^{K} \tilde{\gamma}_{kn}^{\mathbf{z}_{kn}}, \tag{20}$$

where the variational parameter $\tilde{\gamma}_{kn}$ is given by:

$$\tilde{\gamma}_{kn} = \frac{\exp\left\{-\frac{\langle\beta\rangle}{2}\langle\|\mathbf{x}_n - \mathbf{y}_k\|^2\rangle\right\}}{\sum_{k'=1}^{K}\exp\left\{-\frac{\langle\beta\rangle}{2}\langle\|\mathbf{x}_n - \mathbf{y}_{k'}\|^2\rangle\right\}}, \tag{21}$$

where the angled brackets $\langle\cdot\rangle$ denote expectation with respect to the variational distribution $q(\mathbf{Z}, \Theta)$.

2.4.4. *VBM step*

The variational distribution $q(\Theta)$ can be approximated to the product of the variational distribution of each one of the parameters if they are assumed to be i.i.d. If so,

$q(\Theta)$ is expressed as:

$$q(\Theta) = q(\mathbf{Y})q(\beta), \tag{22}$$

where the natural choises of $q(\mathbf{Y})$ and $q(\beta)$ are similar to the priors $p(\mathbf{Y})$ and $p(\beta)$ respectively. Thus,

$$q(\mathbf{Y}) = \prod_{d=1}^{D} \mathcal{N}(\mathbf{y}_{(d)}|\tilde{m}_{(d)}, \tilde{\Sigma}), \tag{23}$$

$$p(\beta) = \Gamma(\beta|\tilde{d}_\beta, \tilde{s}_\beta). \tag{24}$$

Using these expressions in Eq. (13), the formulation for the variational parameters can be obtained:

$$\tilde{\Sigma} = \left(\langle\beta\rangle \sum_{n=1}^{N} \mathbf{G}_n + \mathbf{C}^{-1} \right)^{-1}, \tag{25}$$

$$\tilde{m}_{(d)} = \langle\beta\rangle \tilde{\Sigma} \sum_{n=1}^{N} x_{nd} \langle\mathbf{z}_n\rangle, \tag{26}$$

$$\tilde{d}_\beta = d_\beta + \frac{ND}{2}, \tag{27}$$

$$\tilde{s}_\beta = s_\beta + \frac{1}{2} \sum_{n=1}^{N} \sum_{k=1}^{K} \langle z_{kn}\rangle \langle \|\mathbf{x}_n - \mathbf{y}_k\|^2 \rangle, \tag{28}$$

where $\mathbf{z}_n$ corresponds to each column vector of $\mathbf{Z}$ and $\mathbf{G}_n$ is a diagonal matrix of size $K \times K$ with elements $\langle\mathbf{z}_n\rangle$. The moments in the previous equations are defined as: $\langle z_{kn}\rangle = \tilde{\gamma}_{kn}$, $\langle\beta\rangle = \frac{\tilde{d}_\beta}{\tilde{s}_\beta}$, and $\langle \|\mathbf{x}_n - \mathbf{y}_k\|^2 \rangle = D\tilde{\Sigma}_{kk} + \|\mathbf{x}_n - \tilde{\mathbf{m}}_k\|^2$.

Details of calculations can be found in Ref. 18.

3. Clustering of VBGTM Using FCM

VBGTM produces posterior probabilities for the centres of Gaussian components, but it does not itself provide grouping function based on the latent variables and posterior probabilities. FCM algorithm has grouping function and produces posterior probabilities that indicate the membership of the data points to clusters, but it does not provide visualization if the data dimension is larger than three. While VBGTM is more robust than FCM when processing data set with large variations in probability distributions, we propose to apply an extension to make use of VBGTM technique for fuzzy clustering; we call the extension F-VBGTM.

The goal of F-VBGTM is to train a VBGTM model and use FCM to help VBGTM to cluster the input data into a desired number of clusters. The approach consists of four consecutive steps, like follows:

(1) **Train the VBGTM model**
We train the model as described in Sec. 2.4. The output of the model include the centres of the Gaussian components in the input space (Eq. 26), which can be used as candidate seeds for FCM. The output includes also the posterior probabilities (Eq. 21)

$$\tilde{\gamma}_{kn} = p(k/\mathbf{x}_n), \quad 1 \leq k \leq K, \quad 1 \leq n \leq N,$$

where $\mathbf{x}_n (n = 1, \ldots, N)$ are D-dimensional data vectors.

(2) **Clustering $\tilde{m}_{(d)}$ using FCM**
We apply FCM on the centres of Gaussians $\tilde{m}_{(d)}$ obtained by VBGTM. Suppose there are C clusters. After clustering, the FCM algorithm produces two outputs:

- The cluster seeds: $\nu_c, 1 \leq c \leq C$.
- The membership function for $\tilde{m}_{(d)}$: $p(\nu_c/k)$, $1 \leq k \leq K$, $1 \leq c \leq C$.

(3) **Bayes Theorem**
Now we have the membership of the centres of Gaussians to the cluster seeds ν_c, we must calculate the membership of the original data $\mathbf{x}_n$ to ν_c, we do it using Bayes theorem.

$$\mu_{cn} = p(\nu_c/\mathbf{x}_n) = \sum_{k=1}^{K} p(\nu_c/k) \times p(k/\mathbf{x}_n).$$

(4) **Adjusting**
After Step 3, the original data vectors $\mathbf{x}_n$ are assigned to the new clusters derived from FCM. As a result, the centres have to be adjusted and the distances between data vectors and cluster centres have to be calculated. We use the following equations to do it, $\forall\, c$:

$$\nu_c = \frac{\sum_{n=1}^{N} \mu_{cn} \mathbf{x}_n}{\sum_{n=1}^{N} \mu_{cn}} \quad \text{and} \quad D_{cn} = \|\mathbf{x}_n - \nu_c\|^2.$$

4. Experiments

As explained in the previous section, our hybrid method permits data visualization and grouping at the same time. So we will apply it on several data sets with different size and complexity, then we will compare it with the original FCM to test its performance. The chosen data sets are: Wine, Glass, Iris (all three are available from the UC Irvine (UCI) machine learning repository)[2] and Oil flow data set (available from Netlab package). We will use two internal validity indexes as a criteria to

compare the two methods. Internal validation is based on the information intrinsic to the data alone, without taking into account the real labels. The chosen indexes are: Xie and Beni's index (XB) and Dunn's index (DI) calculated using Fuzzy Clustering and Data Analysis Toolbox[3] for Matlab.

4.1. *Data sets*

- *Wine*: This data set consists of 13 attributes and 179 cases, describing the results of the chemical analysis of samples corresponding to three types of wine.
- *Glass identification*: A data frame with 214 observation containing examples of the chemical analysis of 7 different types of glass. The problem is to forecast the type of class on basis of the chemical analysis. The study of classification of types of glass was motivated by criminological investigation. At the scene of the crime, the glass left can be used as evidence (if it is correctly identified!).
- *Iris*: This data set consists of 50 samples from each of three species of Iris flowers (Iris setosa, Iris virginica and Iris versicolor). Four features were measured from each sample: the length and the width of the sepals and petals, in centimeters.
- *Oil*: This data set consisting of 12 attributes and 1,000 data points was artificially generated from the dynamical equations of a pipeline section carrying a mixture of oil, water and gas, which can belong to one of the three equally distributed geometrical configurations. It was originally used in Ref. 6.

4.2. *Cluster validation*

As a criterion to validate our method and compare it with FCM we use two internal indexes, since internal criteria are used to measure the goodness of a clustering structure without referring to external information (i.e., real labels). We chose two indexes that suit the fuzzy family algorithms. The indexes are the following:

- *Xie and Beni's Index* (XB): This index aims to quantify the ratio of the total variation within clusters and the separation of clusters.[24] A lower value of XB indicates better clustering. It is equal to

$$XB(C) = \frac{\sum_{c=1}^{C} \sum_{n=1}^{N} (\mu_{cn})^m \|\mathbf{x}_n - \nu_c\|^2}{N \times \min_{c,n} \|\mathbf{x}_n - \nu_c\|^2}, \tag{29}$$

  where C is the number of clusters.
- *Dunn's Index* (DI): This index is part of a group of validity indexes including the Davies–Bouldin index, in that it is an internal evaluation scheme. The aim is to identify if clusters are compact, with a small variance between members of the cluster, and well separated. For a given assignment of clusters, a higher DI indicates better clustering.

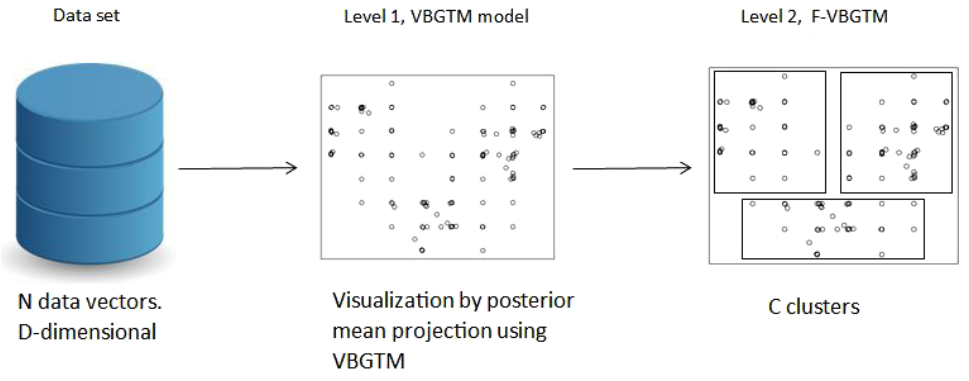


Fig. 2. Illustration of the method, in Level 1 we train a VBGTM model and visualize data in the latent space (2-dimensional) using posterior-mean projection. Then fuzzy clustering of VBGTM in Level 2 to obtain C clusters.

$$DI(C) = \min_{c\in C}\left(\min_{k\in C, k\neq c}\left(\frac{\min_{\mathbf{x}\in C_c,\mathbf{y}\in C_k} d(\mathbf{x},\mathbf{y})}{\max_{k\in C}\{\max_{\mathbf{x},\mathbf{y}\in C} d(\mathbf{x},\mathbf{y})\}}\right)\right). \tag{30}$$

Figures 3 and 4 show the advantage of our method. First, at Level 1, we train a VBGTM model to the data set, we chose 169 latent points (grid of 13×13) for the *Wine* data and 100 (10×10) latent points for *Iris* data. After training the model, we visualize data using the posterior mean projection,[a] results are shown for these two data sets in Figs. 3 and 4. In Fig. 5, we visualize using posterior mode projection.[b] Then, at Level 2, for each VBGTM model we fix a number C of clusters and we apply

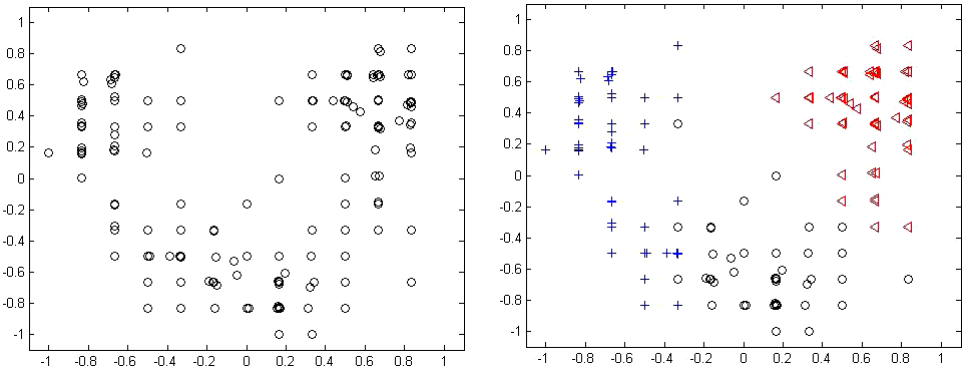
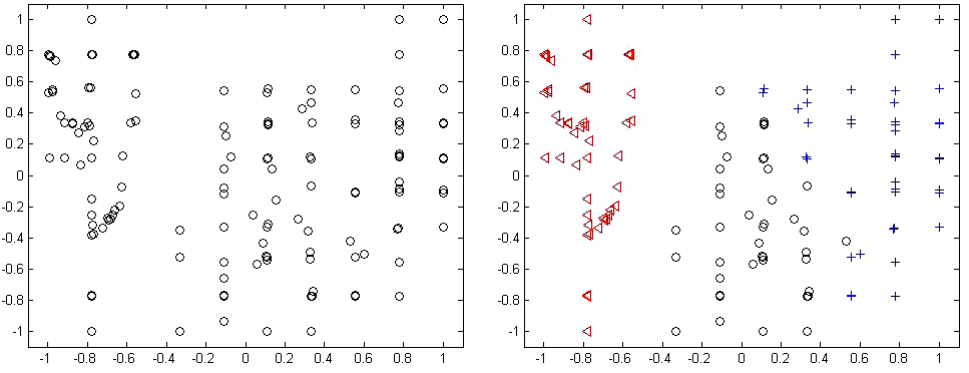


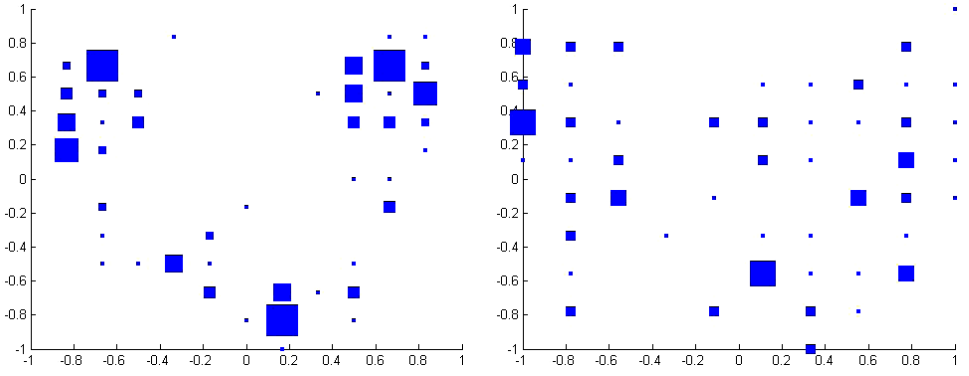
Fig. 3. Visualization of the *Wine* data set using posterior mean projection, with no labels (a) and with labels obtained by applying F-VBGTM (b).

[a] The mean projection is calculated as $z_n^{\text{mean}} = \sum_k \langle z_{kn}\rangle z_k$.
[b] The mode projection is calculated as $k_n^{\text{mode}} = \text{argmax}_k \langle z_{kn}\rangle$.



(a) (b)

Fig. 4. Visualization of the *Iris* data set using posterior mean projection, with no labels (a) and with labels obtained by applying F-VBGTM (b).



(a) (b)

Fig. 5. Visualization of the (a) *Wine* and (b) *Iris* data sets using posterior mode projection. Each square represents a latent point of size proportional to the number of data points assigned to it.

Table 1. Clustering evaluation using *XB* and *DI* for several data sets.

Dataset	Index	FCM	F-VBGTM
Wine	*XB*	0.785	**0.716**
	DI	0.664	0.172
Iris	*XB*	3.794	**2.856**
	DI	0.034	**0.053**
Glass	*XB*	1.131	**0.692**
	DI	0.022	**0.025**
Oil	*XB*	1.594	**1.517**
	DI	0.051	0.048

our method F-VBGTM to group the data, we assign every data point to a cluster (by the highest membership degree) and we visualize the clusters into the same figure obtained by VBGTM's posterior mean projection. Results on data sets *Wine* and *Iris* are shown in Fig. 5(a) and 5(b).

As we mentioned above, a lower value of *XB* indicates better clustering. Table 1 shows that for all the data sets, F-VBGTM has a lower *XB* value than FCM, which shows its better performance based on this index. As for *DI* index, a higher DI indicates better clustering, this is the case for the data sets *Iris* and *Glass*.

5. Collaborative Clustering Using F-VBGTM

Nowadays, computing environments and technologies are more and more evolving towards applications containing massive amounts of heterogeneous, spatially and temporally distributed data sources. Collaborative Clustering[11–13,19] intends to reveal the overall structure of distributed data (i.e., data residing at different repositories) but, at the same time, complying with the restrictions preventing data sharing. Generally speaking, two types of collaborative clustering are envisioned, the horizontal mode and the vertical mode. The vertical mode assumes that each site holds information on different objects described by the same variables, i.e., in the same feature space. The horizontal mode, on the other side, assumes that each location holds information on the same set of objects but described in different feature spaces. The horizontal mode is more complicated since prototypes do not have the same dimension, so defining a distance between them is impossible.

We will give interest for the horizontal case in this paper. Suppose we have P data sets coming from the same population, so the number of objects in each data set is N, the number of Gaussian centres is K. Each data set is referred with an index $[ii], ii = 1, \ldots, P$. The data sets do not have the same variables, so the dimension of the feature space is different from data set to another, let $D[ii]$ be the dimension of the data set $[ii]$, so $D[ii] \neq D[jj]$ if $[ii] \neq [jj]$.

In the horizontal case, as we mentioned above, the collaboration should be done by exchanging the partition matrices between the sites.

Let us suppose that, after training the VBGTM model on the sites $[ii]$ and $[jj]$ then applying the F-VBGTM algorithm on these two sites, the *partition matrix* $U[jj]$ has been sent to the data site $[ii]$. Now we can compute the new cluster prototypes and partition matrix of site $[ii]$, by optimizing the following objective function:

$$J[ii, jj] = \sum_{k=1}^{K} \sum_{i=1}^{C} (u_{ik}[ii])^2 \|\tilde{m}_{(k)} - \nu_i[ii]\|^2 + \beta[ii, jj] \sum_{k=1}^{K} \sum_{i=1}^{C} (u_{ik}[ii] - u_{ik}[jj])^2 \|\tilde{m}_{(k)} - \nu_i[ii]\|^2. \quad (31)$$

The second term in this equation is the term of collaboration, in which the distance between partition matrices is included. By adding this term to the objective function

and optimizing it, the membership degrees are supposed to be closer to each other. The strength of the collaboration is controlled by the coefficient of collaboration $\beta[ii, jj]$, higher $\beta[ii, jj]$ indicates stronger collaboration. Optimizing this objective with respect to $\nu_i[ii]$ and $u_{ik}[ii]$ leads to the following updated equations:

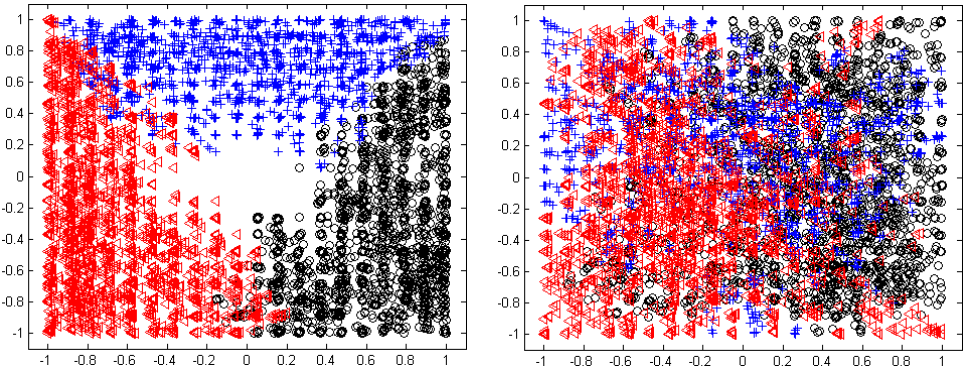
$$u_{ik}[ii] = \frac{1}{\sum_{s=1}^{C} \frac{\|\tilde{m}_{(k)} - \nu_i[ii]\|^2}{\|\tilde{m}_{(k)} - \nu_s[ii]\|^2}} \left[1 - \frac{1}{1 + \beta[ii, jj]} \sum_{s=1}^{C} \beta[ii, jj] u_{sk}[jj]\right] + \frac{\beta[ii, jj] u_{ik}[jj]}{1 + \beta[ii, jj]}, \tag{32}$$

$$\nu_{id}[ii] = \frac{\sum_{t=1}^{K} u_{it}^2[ii] \tilde{m}_{(t)} + \beta[ii, jj] \sum_{t=1}^{K} (u_{it}[ii] - u_{it}[jj])^2 \tilde{m}_{(t)}}{\sum_{k=1}^{K} u_{ik}^2[ii] + \beta[ii, jj] \sum_{k=1}^{K} (u_{ik}[ii] - u_{ik}[jj])^2}, \tag{33}$$

for $i = 1, \ldots, C$, $k = 1, \ldots, K$ and $d = 1, \ldots, D[ii]$.

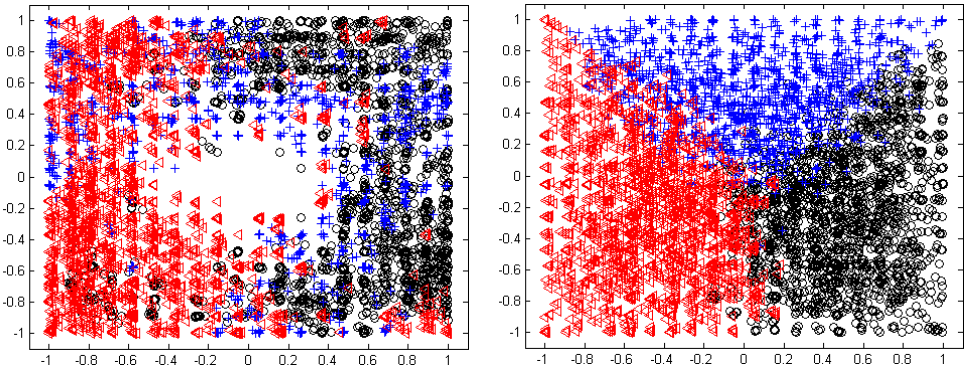
To show the effect of the Collaborative Clustering on the data sites we test the algorithm on a split *Waveform* data set. We choose this data set because of its structure, it contains 21 relevant variables and 19 noisy variables, and 5,000 observations. We split the data set into two subsets, the first subset contains the relevant variables (of dimension 5,000 × 21) and the second subset contains the noisy variables (of dimension 5,000 × 19). By doing this, we get distributed data on two sites, data coming from same population. The first has good clustering results since data are are separable when they are described by the relevant variable. The second site has bad clustering results since its variables are noisy.

The clustering results by F-VBGTM on the two subsets of the *Waveform* data set before collaboration are shown in Fig. 6. Now moving to the collaboration phase, the way we divided the data set, i.e., two subsets with same observations and different variables, permits us to apply the horizontal Collaborative Clustering. We expect that when we



(a) (b)

Fig. 6. Visualization of the two subsets of the *Waveform* data set using posterior mean projection, with labels obtained using F-VBGTM before collaboration. We can see good clustering results on the first subset (a). The results of clustering on the second subset of noisy variables are bad (b).



(a) (b)

Fig. 7. Effect of the Collaborative Clustering. When the second subset (bad) sends results to first subset (good), clusters structure is broken (a). While clustering is ameliorated when the first subset collaborates with the second subset (b).

send the partition matrix of the first subset (relevant) to the second subset (noisy) and then we compute the new prototypes and partition matrix, the results must be better comparing to what we got before the collaboration, because we send results from a good clustered data site to a bad clustered one. The inverse is also true, i.e., when we send the results of the second subset to the first one, this will deteriorate the results of the first subset. The results of the clustering after the collaboration are shown in Fig. 7.

6. Conclusion

In this paper, we developed a new method for fuzzy clustering by combining the VBGTM and the FCM. VBGTM is a variational approximation of GTM, which was proposed as a solution to control data overfitting. Then we used FCM to produce a desired number of clusters based on the output of VBGTM. By combining the two algorithms, we developed a method than can do data visualization and grouping at the same time. Compared to the combination of K-means and SOM, the method proposed in this paper provides membership functions to indicate the likelihood of a data item belonging to a cluster. The membership function is capable of revealing valuable information when performing clustering in applications such as customers segmentation. Experiments showed that the proposed F-VBGTM method consistently performed better than the FCM algorithm. Then we presented a consequence of the proposed algorithm, by applying it to distributed data, more specifically to Collaborative Clustering.

References

1. H. Attias, A variational bayesian framework for graphical models, in *Advances in Neural Information Processing Systems 12* (MIT Press, 2000), pp. 209–215.

2. K. Bache and M. Lichman, UCI Machine Learning Repository. (Available at http://archive.ics.uci.edu/ml).
3. B. Balasko, J. Abonyi and B. Feil, Fuzzy clustering and data analysis toolbox, *Department of Process Engineering, University of Veszprem, Veszprem* (2005).
4. Matthew J. Beal, Variational algorithms for approximate Bayesian inference, Technical report (2003).
5. J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms* (Kluwer Academic Publishers, Norwell, MA, USA, 1981).
6. C. M. Bishop, M. Svensén and C. K. I. Williams, GTM: The Generative Topographic Mapping, *Neural Comput.* **10**(1) (1998) 215–234.
7. C. M. Bishop, M. Svensén and Christopher K. I. Williams, Developments of the generative topographic mapping, *Neurocomputing* **21** (1998) 203–224.
8. H. B. Callen, *Thermodynamics and an Introduction to Thermostatistics*, 2edn. (Wiley, September 1985).
9. A. P. Dempster, N. M. Laird and D. B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, *J. Roy. Stat. Soc. B Met.* **39**(1) (1977) 1–38.
10. Z. Ghahramani and M. J. Beal, Variational inference for bayesian mixtures of factor analysers, in *Advances in Neural Information Processing Systems 12* (MIT Press, 2000), pp. 449–455.
11. M. Ghassany, N. Grozavu and Y. Bennani, Collaborative clustering using prototype-based techniques, *International Journal of Computational Intelligence and Applications* **11**(3) (2012) 1250017.
12. M. Ghassany, N. Grozavu and Y. Bennani, Collaborative Generative Topographic Mapping, in *Neural Information Processing*, *Lecture Notes in Computer Science*, Vol. 7664 (Springer, Berlin, Heidelberg, 2012), pp. 591–598.
13. M. Ghassany, N. Grozavu and Y. Bennani, Collaborative Multi-View Clustering, in *Neural Networks (IJCNN), The 2013 International Joint Conference on*, 2013, pp. 872–879.
14. T. S. Jaakkola and M. I. Jordan, Bayesian parameter estimation via variational methods (1999).
15. T. Kohonen, *Self-Organizing Maps* (Springer-Verlag, Berlin, Berlin, 1995).
16. James MacQueen *et al.*, Some methods for classification and analysis of multivariate observations, in *Proceedings of the Fifth Berkeley Symp. Mathematical Statistics and Probability*, Vol. 1 (California, USA, 1967), p. 14.
17. R. Nock and F. Nielsen, On weighting clustering, *IEEE T. Pattern Anal.* **28**(8) (2006) 1223–1235.
18. I. Olier and A. Vellido, Variational GTM, in *Intelligent Data Engineering and Automated Learning — IDEAL 2007*, Lecture Notes in Computer Science, eds. Hujun Yin, Peter Tino, Emilio Corchado, Will Byrne, and Xin Yao, Vol. 4881, Ch. 9 (Springer, Berlin Heidelberg, Berlin, Heidelberg, 2007), pp. 77–86.
19. W. Pedrycz, Fuzzy clustering with a knowledge-based guidance, *Pattern Recogn. Lett.* **25** (4) (2004) 469–480.
20. E. H. Ruspini, A new approach to clustering, *Information and Control* **15**(1) (1969) 22–32.
21. M. Svensèn, The generative topographic mapping, Technical report, PhD thesis, Aston University (1998).
22. M. A. Tanner, *Tools for Statistical Inference*, 3rd edn. (Springer, 1996).
23. A. Utsugi, Bayesian sampling and ensemble learning in generative topographic mapping, *Neural Process. Lett.* **12**(3) (December 2000) 277–290.
24. X. L. Xie and G. Beni, A validity measure for fuzzy clustering, *IEEE T. Pattern Anal.* **13**(8) (1991) 841–847.