Statistical Models for Partial Membership

Katherine A. Heller HELLER@GATSBY.UCL.AC.UK
Sinead Williamson SAW56@CAM.AC.UK
Zoubin Ghahramani ZOUBIN@ENG.CAM.AC.UK
Engineering Department, University of Cambridge, Cambridge, UK

Abstract

We present a principled Bayesian framework for modeling *partial memberships* of data points to clusters. Unlike a standard mixture model which assumes that each data point belongs to one and only one mixture component, or cluster, a partial membership model allows data points to have fractional membership in multiple clusters. Algorithms which assign data points partial memberships to clusters can be useful for tasks such as clustering genes based on microarray data (Gasch & Eisen, 2002). Our Bayesian Partial Membership Model (BPM) uses exponential family distributions to model each cluster, and a product of these distibtutions, with weighted parameters, to model each datapoint. Here the weights correspond to the degree to which the datapoint belongs to each cluster. All parameters in the BPM are continuous, so we can use Hybrid Monte Carlo to perform inference and learning. We discuss relationships between the BPM and Latent Dirichlet Allocation, Mixed Membership models, Exponential Family PCA, and fuzzy clustering. Lastly, we show some experimental results and discuss nonparametric extensions to our model.

1. Introduction

The idea of *partial membership* is quite intuitive and practically useful. Consider, for example, an individual with a mixed ethnic background, say, partly Asian and partly European. It seems sensible to represent that individual as partly belonging to two different classes or sets. Such a partial membership representation may be relevant to predicting that individual's phenotype, or their food preferences. We clearly need models that can coherently represent partial membership.

Note that partial membership is conceptually very different from uncertain membership. Being certain that a person is partly Asian and partly European, is very different than being uncertain about a person's ethnic background. More information about the person, such as DNA tests, could resolve uncertainty, but cannot make the person change his ethnic membership.

Partial membership is also the cornerstone of fuzzy set theory. While in traditional set theory, items either belong to a set or they don't, fuzzy set theory equips sets with a membership function $\mu_k(x)$ where $0 \leq \mu_k(x) \leq 1$ denotes the degree to which x partially belongs to set k.

In this paper we describe a fully probabilistic approach to data modelling with partial membership. Our approach makes use of a simple way of representing partial membership using continuous latent variables. We define a model which can cluster data but which fundamentally assumes that data points can have partial membership in the clusters. Each cluster is represented by an exponential family distribution with conjugate priors (reviewed in section 3). Our model can be seen as a continuous latent variable relaxation of clustering with finite mixture models, and reduces to mixture modelling under certain settings of the hyperparameters. Unlike Latent Dirichlet Allocation (LDA) (Blei et al., 2003) and Mixed Membership models (Erosheva et al., 2004), which also capture partial membership in the form of attribute-specific mixtures, our model does not assume a factorization over attributes and provides a general way of combining exponential family distributions with partial membership. The complete specification of our model is provided in section 4. Learning and inference are carried out using Markov chain Monte Carlo (MCMC) methods. We show in particular that because all the parameters in

Appearing in *Proceedings of the 25th International Conference on Machine Learning*, Helsinki, Finland, 2008.

our model are continuous, it is possible to employ a full hybrid Monte Carlo (HMC) algorithm, which uses gradients of the log probability, for inference (section 5).

Our Bayesian Partial Membership (BPM) model bears interesting relationships to several well-known models in machine learning and statistics, including LDA (Blei et al., 2003), mixed membership models (Erosheva et al., 2004), exponential family PCA (Collins et al., 2002), and Discrete Components Analysis (Buntine & Jakulin, 2006). We discuss these relations in section 6, where we also relate our model to fuzzy k-means. In section 7, we present both synthetic and real-world experimental results using image data and voting patterns of US senators. We conclude with future work in section 8.

2. A Partial Membership Model

We can derive our method for modeling partial memberships from a standard finite mixture model. In a finite mixture model the probability of a data point, $\boldsymbol{x}_n$ given Θ, which contains the parameters for each of the K mixture components (clusters) is:

$$p(\boldsymbol{x}_n|\Theta) = \sum_{k=1}^{K} \rho_k p_k(\boldsymbol{x}_n|\boldsymbol{\theta}_k) \tag{1}$$

where p_k is the probability distribution of mixture component k, and ρ_k is the mixing proportion (fraction of data points belonging to) for component k[1].

Equation 1 can be rewritten using indicator variables $\boldsymbol{\pi}_n = [\pi_{n1}\pi_{n2}\ldots\pi_{nK}]$ as follows:

$$p(\boldsymbol{x}_n|\Theta) = \sum_{\boldsymbol{\pi}_n} p(\boldsymbol{\pi}_n) \prod_{k=1}^{K} p_k(\boldsymbol{x}_n|\boldsymbol{\theta}_k)^{\pi_{nk}} \tag{2}$$

where $\pi_{nk} \in \{0,1\}$ and $\sum_k \pi_{nk} = 1$. Here we can notice that if $\pi_{nk} = 1$ this means that data point n belongs to cluster k (also $p(\pi_{nk} = 1) = \rho_k$). Therefore the π_{nk} denote *memberships* of data points to clusters.

In order to obtain a model for *partial memberships* we can relax the constraint $\pi_{nk} \in \{0,1\}$ to now allow π_{nk} to take any continuous value in the range $[0,1]$. However, in order to compute the probability of the data under this continuous relaxation of a finite mixture model, we need to modify equation 2 as follows:

$$p(\boldsymbol{x}_n|\Theta) = \int_{\boldsymbol{\pi}_n} p(\boldsymbol{\pi}_n) \frac{1}{c} \prod_{k=1}^{K} p_k(\boldsymbol{x}_n|\boldsymbol{\theta}_k)^{\pi_{nk}} d\boldsymbol{\pi}_n \tag{3}$$

[1]This notation differs slightly from standard notation for mixture models.

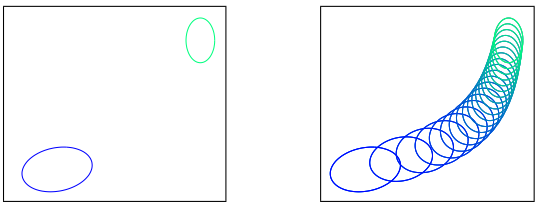


Figure 1. Left: A mixture model with two Gaussian mixture components, or clusters, can generate data from the two distributions shown. Right: Partial membership model with the same two clusters can generate data from all the distributions shown (there are actually infinitely many), which lie between the two original clusters.

The modifications include integrating over all values of $\boldsymbol{\pi}_n$ instead of summing, and since the product over clusters K from equation 2 no longer normalizes we put in a normalizing constant c, which is a function of $\boldsymbol{\pi}_n$ and Θ. Equation 3 now gives us a model for partial membership.

We illustrate the difference between our partial membership model and a standard mixture model in figure 1. Here we can see contours of the Gaussian distributions which can generate data in the mixture model (left) and the partial membership model (right), where both models are using the same two Gaussian clusters. As an example, if one of these clusters represents the ethnicity "White British" and the other cluster represents the ethnicity "Pakistani", then the figure illustrates that the partial membership model will be able to capture someone of mixed ethnicity, whose features may lie in between those of either ethnic group (for example skin color or nose size), better than the mixture model.

3. Conjugate-Exponential Models

In the previous section we derived a partial membership model, given by equation 3. However we have not yet discussed the form of the distribution for each cluster, $p_k(\boldsymbol{x}_n|\boldsymbol{\theta}_k)$, and we will now focus on the case when these distributions are in the exponential family.

An *exponential family distribution* can be written in the form:

$$p_k(\boldsymbol{x}_n|\boldsymbol{\theta}_k) = \exp\{\boldsymbol{s}(\boldsymbol{x}_n)^\top \boldsymbol{\theta}_k + h(\boldsymbol{x}_n) + g(\boldsymbol{\theta}_k)\} \tag{4}$$

where $\boldsymbol{s}(\boldsymbol{x}_n)$ is a vector depending on the data known as the *sufficient statistics*, $\boldsymbol{\theta}_k$ is a vector of *natural parameters*, $h(\boldsymbol{x}_n)$ is a function of the data, and $g(\boldsymbol{\theta}_k)$ is a function of the parameters which ensures that the probability normalizes to one when integrating or summing over $\boldsymbol{x}_n$. We will use the short-hand $\boldsymbol{x}_n \sim \text{Expon}(\boldsymbol{\theta}_k)$ to denote that $\boldsymbol{x}_n$ is drawn from an

exponential family distribution with natural parameters $\boldsymbol{\theta}_k$.

If we plug the exponential family distribution (equation 4) into our partial membership model (equation 3) it follows that:

$$\boldsymbol{x}_n|\boldsymbol{\pi}_n, \Theta \sim \text{Expon}(\sum_k \pi_{nk}\boldsymbol{\theta}_k) \tag{5}$$

where $\boldsymbol{x}_n$ comes from the *same* exponential family distribution as the original clusters p_k, but with *new* natural parameters which are a convex combination of the natural parameters of the original clusters, $\boldsymbol{\theta}_k$, weighted by π_{nk}, the partial membership values for data point $\boldsymbol{x}_n$. Computation of the normalizing constant c is therefore always tractable when p_k is in the exponential family.

A probability distribution $p(\boldsymbol{\theta}_k)$ is said to be *conjugate* to the exponential family distribution $p(\boldsymbol{x}_n|\boldsymbol{\theta}_k)$ if $p(\boldsymbol{\theta}_k|\boldsymbol{x}_n)$ has the same functional form as $p(\boldsymbol{\theta}_k)$. In particular, the conjugate prior to the above exponential family distribution can be written in the form:

$$p(\boldsymbol{\theta}) \propto \exp\{\boldsymbol{\lambda}^\top\boldsymbol{\theta} + \nu g(\boldsymbol{\theta})\} \tag{6}$$

where $\boldsymbol{\lambda}$ and ν are *hyperparameters* of the prior. We will use the short-hand, $\boldsymbol{\theta} \sim \text{Conj}(\boldsymbol{\lambda}, \nu)$. We now have the tools to define our Bayesian partial membership model.

4. Bayesian Partial Membership Models

Consider a model with K clusters, and a data set $\mathcal{D} = \{\boldsymbol{x}_n : n = 1 \dots N\}$. Let $\boldsymbol{\alpha}$ be a K-dimensional vector of positive hyperparameters. We start by drawing mixture weights from a Dirichlet distribution:

$$\boldsymbol{\rho} \sim \text{Dir}(\boldsymbol{\alpha}) \tag{7}$$

Here $\boldsymbol{\rho} \sim \text{Dir}(\boldsymbol{\alpha})$ is shorthand for $p(\boldsymbol{\rho}|\boldsymbol{\alpha}) = c\prod_{k=1}^K \rho_k^{\alpha_k - 1}$ where $c = \Gamma(\sum_k \alpha_k)/\prod_k \Gamma(\alpha_k)$ is a normalization constant which can be expressed in terms of the Gamma function[2]. For each data point, n, we draw a partial membership vector $\boldsymbol{\pi}_n$ which represents how much that data point belongs to each of the K clusters:

$$\boldsymbol{\pi}_n \sim \text{Dir}(a\boldsymbol{\rho}). \tag{8}$$

The parameter a is a positive scaling constant drawn, for example, from an exponential distribution $p(a) = be^{-ba}$, where $b > 0$ is a constant. We assume that each cluster k is characterized by an exponential family distribution with natural parameters $\boldsymbol{\theta}_k$ and that

$$\boldsymbol{\theta}_k \sim \text{Conj}(\boldsymbol{\lambda}, \nu). \tag{9}$$

Given all these latent variables, each data point is drawn from

$$\boldsymbol{x}_n \sim \text{Expon}(\sum_k \pi_{nk}\boldsymbol{\theta}_k) \tag{10}$$

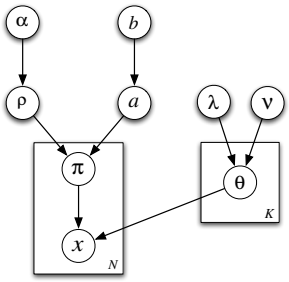


Figure 2. Graphical model for the BPM

In order to get an intuition for what the functions of the parameters we have just defined are, we return to the ethnicity example. Here, each cluster k is an ethnicity (for example, "White British" and "Pakistani") and the parameters $\boldsymbol{\theta}_k$ define a distribution over features for each of the k ethnic groups (for example, how likely it is that someone from that ethnic group likes pizza or marmite or bindi bhaji). The parameter $\boldsymbol{\rho}$ gives the ethnic composition of the population (for example, 75% "White British" and 25% "Pakistani"), while a controls how similar to the population an individual is expected to be (Are 100% of the people themselves 75% "White British" and 25% "Pakistani"? Or are 75% of the people 100% "White British" and the rest are 100% "Pakistani"? Or somewhere in between?). For each person n, $\boldsymbol{\pi}_n$ gives their individual ethnic composition, and finally $\boldsymbol{x}_n$ gives their individual feature values (e.g. how much they like marmite). The graphical model representing this generative process is drawn in Figure 2.

Since the Bayesian Partial Membership Model is a generative model, we tried generating data from it using full-covariance Gaussian clusters. Figure 3 shows the results of generating 3000 data points from our model with $K = 3$ clusters as the value of parameter a changes. We can see that as the value of a increases data points tend to have partial membership in more clusters. In fact we can prove the following lemmas:

Lemma 1 *In the limit that $a \to 0$ the exponential family BPM is a standard mixture model with K components and mixing proportions $\boldsymbol{\rho}$.*

[2]The Gamma function generalizes the factorial to positive reals: $\Gamma(x) = (x-1)\Gamma(x-1)$, $\Gamma(n) = (n-1)!$ for integer n

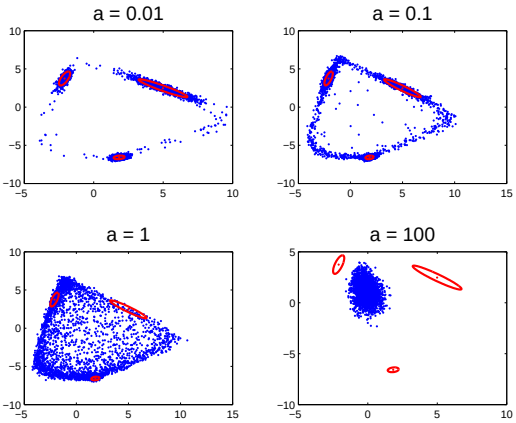


Figure 3. 3000 BPM generated data points with partial assignments to 3 Gaussian clusters shown in red, as parameter a varies.

Lemma 2 *In the limit that $a \to \infty$ the exponential family BPM model has a single component with natural parameters $\sum_k \rho_k \boldsymbol{\theta}_k$.*

Proofs of these lemmas follow simply from taking the limits of equation 8 as a goes to 0 and ∞ respectively.

5. BPM Learning

We can represent the observed data set $\mathcal{D}$ as an $N \times D$ matrix $\mathbf{X}$ with rows corresponding to $\boldsymbol{x}_n$, where D is the number of input features.[3] Let $\boldsymbol{\Theta}$ be a $K \times D$ matrix with rows $\boldsymbol{\theta}_k$ and $\boldsymbol{\Pi}$ be an $N \times K$ matrix with rows $\boldsymbol{\pi}_n$. Learning in the BPM consists of inferring all unknown variables, $\boldsymbol{\Omega} = \{\boldsymbol{\Pi}, \boldsymbol{\Theta}, \boldsymbol{\rho}, a\}$ given $\mathbf{X}$. We treat the top level variables in the graphical model in Figure 2, $\boldsymbol{\Psi} = \{\boldsymbol{\alpha}, \boldsymbol{\lambda}, \nu, b\}$ as fixed hyperparameters, although these could also be learned from data. Our goal is to infer $p(\boldsymbol{\Omega}|\mathbf{X}, \boldsymbol{\Psi})$, for which we decide to employ Markov chain Monte Carlo (MCMC).

Our key observation for MCMC is that even though BPMs contain discrete mixture models as a special case, *all* of the unknown variables $\boldsymbol{\Omega}$ of the BPM are continuous. Moreover, it is possible to take derivatives of the log of the joint probability of all variables with respect to $\boldsymbol{\Omega}$. This makes it possible to do inference using a full Hybrid Monte Carlo (HMC) algorithm on all parameters. Hybrid (or Hamiltonian) Monte Carlo is an MCMC procedure which overcomes the random walk behaviour of more traditional Metropolis or Gibbs sampling algorithms by making use of the derivatives of the log probability (Neal, 1993; MacKay, 2003). In high dimensions, this derivative information can lead to a dramatically faster mixing of the Markov chain, analogous to how optimization using derivatives is often much faster than using greedy random search.

We start by writing the probability of all parameters and variables[4] in our model:

$$p(\mathbf{X}, \boldsymbol{\Omega}|\boldsymbol{\Psi}) = p(\mathbf{X}|\boldsymbol{\Pi}, \boldsymbol{\Theta}) p(\boldsymbol{\Theta}|\boldsymbol{\lambda}, \nu) p(\boldsymbol{\Pi}|a, \boldsymbol{\rho}) p(a|b) p(\boldsymbol{\rho}|\boldsymbol{\alpha}) \tag{11}$$

We assume that the hyperparameter $\nu = 1$, and omit it from our derivation. Since the forms of all distributions on the right side of equation (11) are given in section 4, we can simply plug these in and see that:

$$\begin{aligned}
&\log p(\mathbf{X}, \boldsymbol{\Omega}|\boldsymbol{\Psi}) = \\
&\quad \log \Gamma(\textstyle\sum_k \alpha_k) - \sum_k \log \Gamma(\alpha_k) + \sum_k (\alpha_k - 1) \log \rho_k \\
&\quad + \log b - ba + N \log \Gamma\left(\textstyle\sum_k a\rho_k\right) - N \sum_k \log \Gamma(a\rho_k) \\
&\quad + \textstyle\sum_n \sum_k (a\rho_k - 1) \log \pi_{nk} + \sum_k \left[\boldsymbol{\theta}_k^\top \boldsymbol{\lambda} + \mathrm{g}(\boldsymbol{\theta}_k) + \mathrm{f}(\boldsymbol{\lambda})\right] \\
&\quad + \textstyle\sum_n \left[\left(\sum_k \pi_{nk} \boldsymbol{\theta}_k\right)^\top \boldsymbol{x}_n + \mathrm{h}(\boldsymbol{x}_n) + \mathrm{g}\left(\sum_k \pi_{nk} \boldsymbol{\theta}_k\right)\right]
\end{aligned}$$

The Hybrid Monte Carlo algorithm simulates dynamics of a system with continuous state $\boldsymbol{\Omega}$ on an energy function $\mathcal{E}(\boldsymbol{\Omega}) = -\log p(\mathbf{X}, \boldsymbol{\Omega}|\boldsymbol{\Psi})$. The derivatives of the energy function $\frac{\partial \mathcal{E}(\boldsymbol{\Omega})}{\partial \boldsymbol{\Omega})}$ provide forces on the state variables which encourage the system to find high probability regions, while maintaining detailed balance to ensure that the correct equilibrium distribution over states is achieved (Neal, 1993). Since $\boldsymbol{\Omega}$ has constraints, e.g. $a > 0$ and $\sum_k \rho_k = 1$, we use a tranformation of variables so that the new state variables are unconstrained, and we perform dynamics in this unconstrained space. Specifically, we use $a = e^\eta$, $\rho_k = \frac{e^{r_k}}{\sum_{k'} e^{r_{k'}}}$, and $\pi_{nk} = \frac{e^{p_{nk}}}{\sum_{k'} e^{p_{nk'}}}$. For HMC to be valid in this new space, the chain rule needs to be applied to the derivatives of $\mathcal{E}$, and the prior needs to be transformed through the Jacobian of the change of variables. For example, $p(a)da = p(\eta)d\eta$ implies $p(\eta) = p(a)(da/d\eta) = ap(a)$. We also extended the HMC procedure to handle missing inputs in a principled manner, by analytically integrating them out, as this was required for some of our applications. More details and general pseudocode for HMC can be found in (MacKay, 2003).

6. Related Work

The BPM model has interesting relations to several models that have been proposed in machine learning, statistics and pattern recognition. We describe these relationships here.

[3] We assume that the data is represented in its natural representation for the exponential family likelihood, so that $\boldsymbol{s}(\boldsymbol{x}_n) = \boldsymbol{x}_n$.

[4] A formal distinction between hidden variables, e.g. the $\{\boldsymbol{\pi}_n\}$, and unknown parameters is not necessary as they are both unknowns.

Latent Dirichlet Allocation: Using the notation introduced above, the BPM model and LDA (Blei et al., 2003) both incorporate a K-dimensional Dirichlet distributed $\boldsymbol{\pi}$ variable. In LDA, $\boldsymbol{\pi}_n$ are the mixing proportions of the topic mixture for each document n. Each word in document n can then be seen as having been generated by topic k, with probability π_{nk}, where the word distribution for topic k is given by a multinomial distribution with some parameters, $\boldsymbol{\theta}_k$. The BPM also combines π_{nk} with some exponential family parameters $\boldsymbol{\theta}_k$, but here the way in which they are combined does not result in a mixture model from which another variable (e.g. a word) is assumed to be generated. In contrast, the data points are indexed by n directly, and therefore exist at the document level of LDA. Each data point is assumed to have come from an exponential family distribution parameterized by a weighted sum of natural parameters $\boldsymbol{\theta}$, where the weights are given by π_n for data point n. In LDA, data is organized at two levels (e.g. documents and words). More generally, mixed membership (MM) models (Erosheva et al., 2004), or admixture models, assume that each data attribute (e.g. words) of the data point (e.g. document) is drawn independently from a mixture distribution given the membership vector for the data point, $x_{nd} \sim \sum_k \pi_{nk} P(x|\theta_{kd})$. LDA and mixed membership models do not average natural parameters of exponential family distributions like the BPM. LDA or MM models could not generate the continuous densities in figure 3 from full-covariance Gaussians. The analagous generative process for MM models is given in figure 4. Since data attributes are drawn independently, the original clusters (not explicity shown) are one dimensional and have means at 0, 10 and 20 for both attribute dimensions. We can notice from the plot that this model always generates a mixture of 9 Gaussians, which is a very different behavior than the BPM, and clearly not as suitable for the general modeling of partial memberships. LDA only makes sense when the objects (e.g. documents) being modelled constitute bags of exchangeable sub-objects (e.g. words). Our model makes no such assumption. Moreover, in LDA and MM models there is a discrete latent variable for every sub-object corresponding to which mixture component that sub-object was drawn from. This large number of discrete latent variables makes MCMC sampling in LDA potentially much more expensive than in BPM models.

Exponential Family PCA: Our model bears an interesting relationship to Exponential Family PCA (Collins et al., 2002). EPCA was originally formulated as the solution to an optimization problem based on Bregman divergences, while our model is a fully probabilistic model in which all parameters can be integrated out via MCMC. However, it is possible to think of EPCA as the likelihood function of a probabilistic model, which coupled with a prior on the parameters, would make it possible to do Bayesian inference in EPCA and would render it closer to our model. However, our model was entirely motivated by the idea of partial membership in clusters, which is enforced by forming convex combinations of the natural parameters of exponential family models, while EPCA is based on *linear* combinations of the parameters. Therefore: EPCA does not naturally reduce to clustering, none of the variables can be interpreted as partial memberships, and the coefficients define a plane rather than a convex region in parameter space.

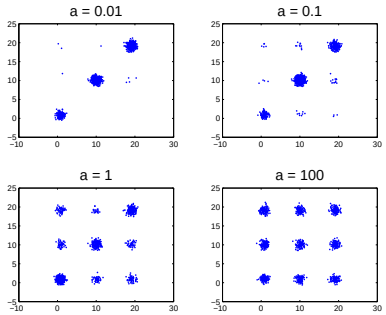


Figure 4. Generative plot for MM model with 3 Gaussian clusters

The recent work of Buntine and Jakulin (Buntine & Jakulin, 2006) focusing on the analysis of discrete data is also closely related to the BPM model. The framework of (Buntine & Jakulin, 2006) section III B expresses a model for discrete data in terms of linear mixtures of dual exponential family parameters where MAP inference is performed. Section V B also provides insights on differences between using dual and natural parameters.

Fuzzy Clustering: The notion that probabilistic models are unable to handle partial membership has been used to argue that probability is a subtheory of or different in character from fuzzy logic (Zadeh, 1965; Kosko, 1992). In this paper we described a probabilistic model for partial membership which may be of use in the many application domains where fuzzy clustering has been used.

Fuzzy K-means clustering (Bezdek, 1981) iteratively minimizes the following objective: $J = \sum_{n=1}^{N}\sum_{k=1}^{K} \pi_{nk}^{\gamma} d^2(\boldsymbol{x}_n, \boldsymbol{c}_k)$, where $\gamma > 1$ is an exponent parameter, π_{nk} represents the degree of membership of

data point n in cluster k ($\sum_k \pi_{nk} = 1$), and $d^2(\boldsymbol{x}_n, \boldsymbol{c}_k)$ is a measure of squared distance between data point $\boldsymbol{x}_n$ and cluster center $\boldsymbol{c}_k$. By varying γ it is possible to attain different amounts of partial membership, where the limiting case $\gamma = 1$ is K-means with no partial membership. Although the π parameters represent partial membership, none of the variables have probabilistic interpretations.

IOMM: Lastly, this work is related to the Infinite Overlapping Mixture Model (IOMM) (Heller & Ghahramani, 2007) in which overlapping clustering is performed, also by taking products of exponential family distributions, much like products of experts (Hinton, 1999). However in the IOMM the memberships of data points to clusters are restricted to be binary, which means that it can not model partial membership.

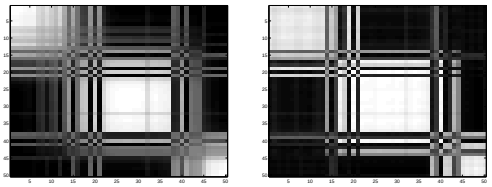
7. Experiments

We generated a synthetic binary data set from the BPM, and used this to test BPM learning. The synthetic data set had 50 data points which each have 32 dimensions and can hold partial memberships in 3 clusters. We ran our Hybrid Monte Carlo sampler for 4000 iterations, burning in the first half. In order to compare our learned partial membership assignments for data points (Π_L) to the true ones (Π_T) for this synthetic data set, we compute ($\hat{U} = \Pi_L \Pi_L^\top$) and ($U^* = \Pi_T \Pi_T^\top$), which basically give the total amount of cluster membership shared between each pair of data points, and is invariant to permutations of cluster labels. Both of these matrices can be seen in figure 5. One can see that the structure of these two matrices is quite similar, and that the BPM is learning the synthetic data reasonably. For a more quantitative measure table 5c gives statistics on the number of pairs of data points whose learned shared membership differs from the true shared membership by more than a given threshold (the range of this statistic is [0,1]).

We also used the BPM to model two "real-world" data sets. The first is senate roll call data from the 107th US congress (2001-2002) (Jakulin, 2004), and the second is a data set of images of sunsets and towers.

The senate roll call data is a matrix of 99 senators (one senator died in 2002 and neither he nor his replacement is included) by 633 votes. It also includes the outcome of each vote, which is treated as an additional data point (like a senator who always voted the actual outcome). The matrix contained binary features for yea and nay votes, and we used the BPM to cluster this data set using $K = 2$ clusters. There are missing values in this dataset but this can easily be dealt with in the HMC log probability calculations by explicitly representing both 0 and 1 binary values and leaving out missing values. The results are given in figure 6. The line in figure 6 represents the amount of membership of each senator in one of the clusters (we used the "Democrat" cluster, where senators on the far left have partial memberships very close to 0, and those on the far right have partial memberships extremely close to 1). Since there are two clusters, and the amount of membership always sums to 1 across clusters, the figure looks the same regardless of whether we are looking at the "Democrat" or "Republican" cluster. We can see that most Republicans and Democrats are tightly clustered at the ends of the line (and have partial memberships very close to 0 and 1), but that there is a fraction of senators (around 20%) which lies somewhere reasonably in between the extreme partial memberships of 0 or 1. Interesting properties of this figure include the location of Senator Jeffords who left the Republican party in 2001 to become an independent who caucused with the Democrats. Also Senator Chafee who is known as a moderate Republican and who often voted with the Democrats (for example, he was the only Republican to vote against authorizing the use of force in Iraq), and Senator Miller a conservative Democrat who supported George Bush over John Kerry in the 2004 US Presidential elections. Lastly, it is interesting to note the location of the Outcome data point, which is very much in the middle. This makes sense since the 107th congress was split 50-50 (with Republican Dick Cheney breaking ties), until Senator Jeffords became an Independent at which point the Democrats had a one seat majority.

We also tried running both fuzzy k-means clustering and Dirichlet Process Mixture models (DPMs) on this data set. While fuzzy k-means found roughly similar rankings of the senators in terms of membership to the "Democrat" cluster, the exact ranking and, in particular, the amount of partial membership ($\pi_{\mathbf{n}}$) each senator had in the cluster was *very* sensitive to the fuzzy exponent parameter, which is typically set by hand. Figure 7a plots the amount of membership for the Outcome data point in black, as well as the most extreme Republican, Senator Ensign, in red, and the most extreme Democrat, Senator Schumer, in blue, as a function of the fuzzy exponent parameter. We can see in this plot that as the assignment of the Outcome data point begins to reach a value even reasonably close to 0.5, the most extreme Republican already has 20% membership in the "Democrat" cluster. This reduction in range does not make sense semantically, and presents a trade-off between finding reasonable values



Statistic	Percent
$\lvert(\hat{U} - U^*)\rvert \leq 0.1$	60.40
$\lvert(\hat{U} - U^*)\rvert \leq 0.2$	84.28
$\lvert(\hat{U} - U^*)\rvert \leq 0.3$	95.48
$\lvert(\hat{U} - U^*)\rvert \leq 0.4$	99.68
$\lvert(\hat{U} - U^*)\rvert \leq 0.5$	100.00

Figure 5. a) left - matrix U^* showing the true shared partial memberships for pairs of data points. b) right - matrix $\hat{U}$ showing the learned shared partial memberships. c) Summary statistics for learned $\hat{U}$. Reports the percentage of pairs in $\hat{U}$ whose difference from U^* in terms of the amount of shared partial memberships is at most the given threshold (0.1 - 0.5).

for π_n in the middle of the range, versus at the extremes. This kind of sensitivity to parameters does not exist in our BPM model, which models both extreme and middle range values well.

We tried using a DPM to model this data set where we ran the DPM for 1000 iterations of Gibbs sampling, sampling both assignments and concentration parameter. The DPM confidently finds 4 clusters: one cluster consists solely of Democrats, one consists solely of Republicans, the third cluster has 9 of the most moderate Democrats and Republicans plus the "vote outcome" variable, and the last cluster has just one member, Hollings (D-SC). Figure 7b is a 100x100 matrix showing the overlap of cluster assignments for pairs of senators, averaged over 500 samples (there are no changes in relative assignments, the DPM is completely confident). The interpretation of the data provided by the DPM is very different from the BPM model's. The DPM does *not* use uncertainty in cluster membership to model Senators with intermediate views. Rather, it creates an entirely new cluster to model these Senators. This makes sense for the data as viewed by the DPM: there is ample data in the roll calls that these Senators are moderate — it is not the case that there is uncertainty about whether they fall in line with hardcore Democrats or Republicans. This highlights the fact that the responsibilities in a mixture model (such as the DPM) cannot and should not be interpreted as partial membership, they are representations of *uncertainty* in full membership. The BPM model, however, explicitly models the partial membership, and can, for example, represent the fact that a Senator might be best characterized as moderate (and quantify how moderate they are). In order to quantify this comparison we calculated the negative log predictive probability (in bits) across senators for the BPM and the DPM (Table 1). We look at a number of different measures: the mean, median, minimum and maximum number of bits required to encode a senator's votes. We also look at the number of bits needed to encode the "Outcome" in particular. On all of these measures

	Mean	Median	Min	Max	"Outcome"
BPM	**187**	**168**	**93**	422	**224**
DPM	196	178	112	**412**	245

Table 1. Comparison between the BPM and a DPM in terms of negative log predictive probability (in bits) across senators.

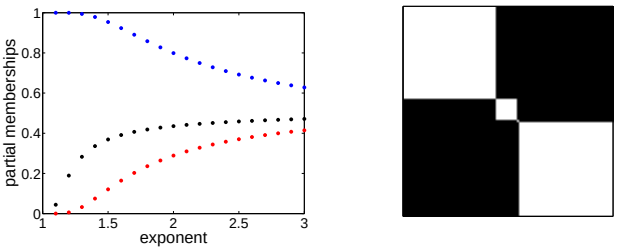


Figure 7. a) left - fuzzy k-means: plot of the partial membership values for the Outcome data point (in black) and the most extreme Republican (in red) and Democrat (in blue) as a function of the fuzzy exponent parameter. b) right - DPMs: an ordered 100x100 matrix showing the fraction of times each pair of senators was assigned to the same cluster, averaged over 500 Gibbs sampling iterations.

except for maximum, the BPM performs better than the DPM, showing that the BPM is a superior model for this data set.

Lastly, we used the BPM to model images of sunsets and towers. The dataset consisted of 329 images of sunsets or towers, each of which was represented by 240 binary simple texture and color features. Partial assignments to $K = 2$ clusters were learned, and figure 8 provides the result. The top row of the figure is the three images with the most membership in the "sunset" cluster, the bottom row contains the three images with the most membership in the "tower" cluster, and the middle row shows the 3 images which have closest to 50/50 membership in each cluster ($\pi_{nk} \approx 0.5$). In this dataset, as well as all the datasets described in this section, our HMC sampler was very fast, giving reasonable results within tens of seconds.

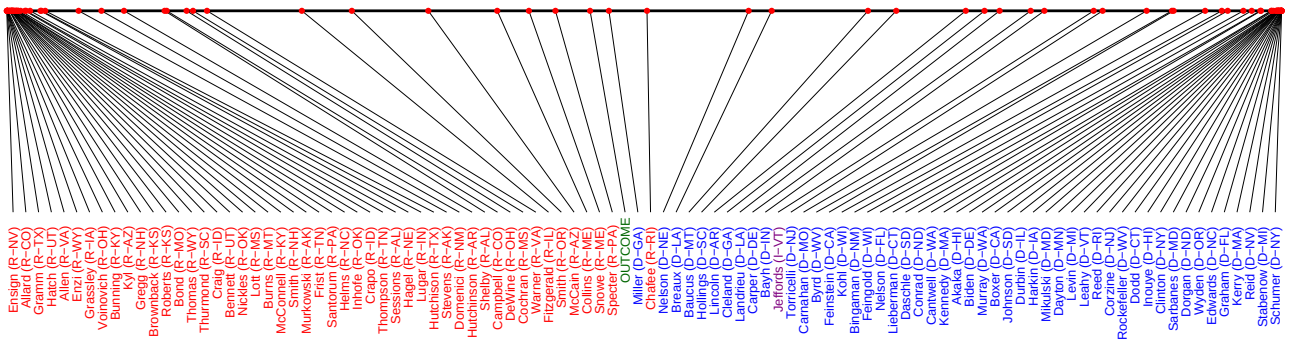


Figure 6. Analysis of the partial membership results on the Senate roll call data from 2001-2002. The line shows amount of membership in the "Democrat" cluster with the left of the line being the lowest and the right the highest.

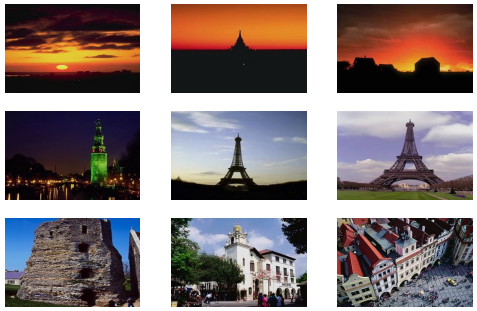


Figure 8. Tower and Sunset images. The top row are the images found to have largest membership in the "sunset" cluster, the bottom row are images found to have largest membership in the "tower" cluster, and the middle row are the images which have the most even membership in both clusters.

8. Conclusions and Future Work

In summary, we have described a fully probabilistic approach to data modelling with partial membership using continuous latent variables, which can be seen as a relaxation of clustering with finite mixture models. We employed a full Hybrid Monte Carlo algorithm for inference, and our experience with HMC has been very positive. Despite the general reputation of MCMC methods for being slow, our model using HMC seems to discover sensible partial membership structure after surprisingly few samples.

In the future we would like to develop a nonparametric version of this model. The most obvious way to try to generalize this model would be with a Hierarchical Dirichlet Process (Teh et al., 2006). However, this would involve averaging over infinitely many potential clusters, which is both computationally infeasible, and also undesirable from the point of view that each data point should have non-zero partial membership in only a few (certainly finite) number of clusters. A more promising alternative is to use an Indian Buffet Process (Griffiths & Ghahramani, 2005), where each 1 in a row in an IBP sample matrix would represent a cluster in which the data point corresponding to that row has non-zero partial membership, and then draw the continuous values for those partial memberships conditioned on that IBP matrix.

References

Bezdek, J. (1981). *Pattern recognition with fuzzy objective function algorithms*. Kluwer.

Blei, D., Ng, A., & Jordan, M. (2003). Latent dirichlet allocation. *JMLR*.

Buntine, W., & Jakulin, A. (2006). *LNCS*, vol. 3940, chapter Discrete Component Analysis. Springer.

Collins, M., Dasgupta, S., & Schapire, R. (2002). A generalization of principal components analysis to the exponential family. *NIPS*.

Erosheva, E., Fienberg, S., & Lafferty, J. (2004). Mixed membership models of scientific publications. *PNAS*.

Gasch, A., & Eisen, M. (2002). Exploring the conditional coregulation of yeast gene expression through fuzzy k-means clustering. *Genome Biol.*, *3*.

Griffiths, T., & Ghahramani, Z. (2005). *Infinite latent feature models and the indian buffet process* (Technical Report). Gatsby Computational Neuroscience Unit.

Heller, K., & Ghahramani, Z. (2007). A nonparametric bayesian approach to modeling overlapping clusters. *AISTATS*.

Hinton, G. (1999). Products of experts. *ICANN*.

Jakulin, A. (2004). http://www.ailab.si/aleks/politics/.

Kosko, B. (1992). *Neural networks and fuzzy systems*. Prentice Hall.

MacKay, D. (2003). *Information theory, inference, and learning algorithms*. Cambridge University Press.

Neal, R. (1993). *Probabilistic inference using markov chain monte carlo methods* (Technical Report). University of Toronto.

Teh, Y., Jordan, M., Beal, M., & Blei, D. (2006). Hierarchical dirichlet processes. *JASA*, *101*.

Zadeh, L. (1965). Fuzzy sets. *Info. and Control*, *8*.