Optimal Bayesian estimators for latent variable cluster models

Riccardo Rastelli[1,2,*] and Nial Friel[1,2]

[1]School of Mathematics and Statistics, University College Dublin, Ireland;
[2]Insight: Centre for Data Analytics, Ireland.

March 23, 2017

Abstract

In cluster analysis interest lies in probabilistically capturing partitions of individuals, items or observations into groups, such that those belonging to the same group share similar attributes or relational profiles. Bayesian posterior samples for the latent allocation variables can be effectively obtained in a wide range of clustering models, including finite mixtures, infinite mixtures, hidden Markov models and block models for networks. However, due to the categorical nature of the clustering variables and the lack of scalable algorithms, summary tools that can interpret such samples are not available. We adopt a Bayesian decision theoretic approach to define an optimality criterion for clusterings, and propose a fast and context-independent greedy algorithm to find the best allocations. One important facet of our approach is that the optimal number of groups is automatically selected, thereby solving the clustering and the model-choice problems at the same time. We consider several loss functions to compare partitions, and show that our approach can accommodate a wide range of cases. Finally, we illustrate our approach on a variety of real-data applications for three different clustering models: Gaussian finite mixtures, stochastic block models and latent block models for networks.

Keywords: Bayesian clustering, Cluster analysis, Greedy optimisation, Latent variable models, Markov chain Monte Carlo.

1   Introduction

Cluster analysis plays a central role in statistics and machine learning, yet it is not immediately clear how one can appropriately summarise the output of partitions from a Bayesian clustering model. This article seeks to address this impasse, proposing an optimality criterion for clusterings derived from decision theory, and a greedy algorithm to estimate the optimal partition and number of groups. Clustering models are often represented as discrete latent variable models: each of the data objects corresponds to the elements of $\mathcal{V} = \{1, 2, \dots, N\}$ and is characterised by a categorical latent variable

$z = \{1, 2, \ldots, K\}$ denoting its group label. Such variables are often called *clustering variables* or *allocations*. Notable examples of latent variable clustering models include: product partition models (Hartigan 1990; Barry and Hartigan 1992), finite mixtures (McLachlan and Peel 2004), infinite mixtures (Quintana (2006) and references therein), latent block models for networks (Nowicki and Snijders 2001; Govaert 1995), hidden Markov models (MacDonald and Zucchini 1997).

The motivation for this paper ensues from the introduction within the statistical community of the so-called trans-dimensional samplers. One well known and widely used sampler is the *reversible jump algorithm* of Green (1995), extended to the context of finite mixtures by Richardson and Green (1997) and to hidden Markov models by Robert et al. (2000). Reversible jump Markov chain Monte Carlo allows one to explore a number of models with a single Markov chain that "jumps" between them, thereby estimating both the model parameters and the posterior model probabilities. A more recent trans-dimensional Markov Chain Monte Carlo algorithm is the *allocation sampler* introduced by Nobile and Fearnside (2007). This takes advantage of the fact that, in some finite mixture models, the marginal posterior distribution of the allocation variables can be obtained by analytically integrating out all of the model parameters. This allows one to use a *collapsed* Gibbs sampler and obtain a posterior marginal sample for the clustering variables. One advantage of this method is that the number of groups can be inferred at each step from the clustering variables automatically, hence obtaining posterior probabilities for the different models. The core idea of the allocation sampler has been recently extended to a number of frameworks, including latent class analysis (White et al. 2016), latent block models (Wyse and Friel 2012), stochastic block models (McDaid et al. 2013), latent position models (Friel et al. 2013), and change point analysis (Benson and Friel 2016). In Bayesian nonparametrics a similar approach has been proposed by (Neal 2000; Favaro and Teh 2013) for Dirichlet process mixture models.

Both reversible jump and allocation sampler return a trans-dimensional sample for the allocations. Theoretically, such a sample contains all of the posterior information needed for the clustering of the data, however, interpreting such information is a very challenging task. Since the allocations are categorical variables, usual summary statistics such as the mean, median and quantiles are not well defined. In addition, these Markov chain Monte Carlo algorithms are sensitive to label-switching issues (Stephens 2000), in fact, when using the latent variable representation, all mixture models are non-identifiable up to a permutation of the cluster labels. In addition, the sample itself may be computationally impractical to handle, since even basic operations may require a cost that grows with N^2 or the square of the size of the sample.

The problem described really boils down to a very simple research question: we want to summarise the information provided by a sample of partitions into an optimal partition. This issue has been addressed in several previous works, such as Strehl and Ghosh

(2003), Gionis et al. (2007), Dahl (2009), and Fritsch and Ickstadt (2009), where the authors propose a number of approaches that define a theoretical optimal partition and introduce algorithms to find it. One critique to these contributions is that the proposed methodologies lack a sound theoretical background and they may be seen as ad-hoc.

In this work we use a Bayesian decision theoretic framework to define an optimality criterion for partitions, as previously proposed by Binder (1978), Lau and Green (2007), and Wade and Ghahramani (2015). From the Bayesian theoretical point of view, our approach defines the best possible solution to the partitioning problem using the information contained in the sample. Also, an important facet of this methodology is that it builds upon recent adaptations of the allocation sampler (Wyse and Friel 2012; McDaid et al. 2013; Friel et al. 2013; White et al. 2016), making up for one important shortcoming of these samplers: the interpretation of the results.

The essence of the decision theoretic framework lies in the definition of a loss function in the space of partitions, which is often a metric measuring how different two partitions are. Then, the optimal partition is estimated as the one minimising the average loss with respect to the sample given. In the Bayesian perspective, this is equivalent to adopting a Bayes estimator (or *Bayes action*), which is the decision minimising the *Expected Posterior Loss* (EPL).

We propose a greedy algorithm as means to find the optimal partition, focusing on its computational complexity and scalability. The algorithm can deal with a wide family of loss functions and requires only the sample of partitions as input. Hence our methodology has wide applicability and is the only scalable procedure that can be used to perform Bayesian clustering for a relatively arbitrary loss function. One important advantage of our algorithmic frameworks is that the resulting optimal clustering automatically determines the optimal number of groups.

Previous works (Lau and Green 2007; Wade and Ghahramani 2015) were confined to the case of Bayesian nonparametric models. Here we stress that this approach is automatically extended to a very general clustering context, and hence we propose applications to several different frameworks.

The plan of the paper is summarised as follows: Section 2 describes the theoretical foundations of Bayesian clustering; in Section 3 we describe the properties of several loss functions to compare partitions, and we characterise the wide breadth to which our method extends; in Section 4 we introduce our greedy algorithm and analyse its complexity and features, whereas Section 5 shows an interesting procedure that can be used to potentially save an amount of computational time. Finally, three applications to real datasets are proposed in Section 6: the galaxies' dataset for univariate Gaussian finite mixtures, the French political blogosphere for stochastic block models, and the congressional voting data for latent block models. Section 7 closes the paper with some final comments.

2 Bayesian clustering: the theory

Let $\mathbf{Z}$ be a $T \times N$ matrix, where, for every $t = 1, \ldots, T$ and $i = 1, \ldots, N$, z_{ti} is a categorical variable (typically $z_{ti} \in \{1, 2, \ldots, N\}$) indicating the cluster label of observation i at iteration t. The rows of $\mathbf{Z}$ determine a sample of partitions of the same set $\mathcal{V} = \{1, 2, \ldots, N\}$, and we assume that such sample is drawn from the posterior distribution of a clustering model, given the observed data $\mathcal{Y}$. An alternative representation of the sample would be $\left\{\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(T)}\right\}$, where $\mathbf{z}^{(t)} = \{z_{t1}, \ldots, z_{tN}\} \in \mathcal{Z}$ corresponds to the t-th row of $\mathbf{Z}$, and $\mathcal{Z}$ is the space of all partitions of $\mathcal{V}$.

Interest lies in conveying the information provided by the posterior sample into a single optimal partition. Bayesian decision theory offers an elegant approach to tackle this task, essentially recasting the clustering problem into one of decision making.

The first step consists of choosing a loss function $\mathcal{L} : \mathcal{Z} \times \mathcal{Z} \to \mathbb{R}$. For any two partitions (hereafter also called *decisions*) $\mathbf{a}$ and $\mathbf{z}$, the quantity $\mathcal{L}(\mathbf{a}, \mathbf{z})$ indicates the loss occurring when the decision $\mathbf{a}$ is chosen while $\mathbf{z}$ is the correct partition. The choice of the loss function adopted is completely arbitrary and supposedly situational, nonetheless some loss functions have interesting features and tend to work well in many contexts. A loss function is not necessarily a distance in the space of partitions although this is often regarded as a desirable property, since it helps particularly in the interpretation and representation of the results.

An optimal decision (also called *Bayes action*) is one minimising the *expected posterior loss*, defined as:

$$\Psi(\mathbf{a}) := \mathbb{E}_{\mathbf{z}}\left[\mathcal{L}(\mathbf{a}, \mathbf{z})|\mathcal{Y}\right] = \sum_{\mathbf{z} \in \mathcal{Z}} \pi(\mathbf{z}|\mathcal{Y}) \mathcal{L}(\mathbf{a}, \mathbf{z}). \tag{1}$$

Considering that the posterior sample $\left\{\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(T)}\right\} \sim \pi(\cdot|\mathcal{Y})$ is available, for every decision $\mathbf{a} \in \mathcal{Z}$, an unbiased estimator of the associated expected posterior loss results as:

$$\psi(\mathbf{a}) = \frac{1}{T} \sum_{t=1}^{T} \mathcal{L}\left(\mathbf{a}, \mathbf{z}^{(t)}\right) \approx \Psi(\mathbf{a}). \tag{2}$$

We aim then at finding the decision $\hat{\mathbf{a}}$ minimising the approximate expected posterior loss:

$$\hat{\mathbf{a}} = \arg\min_{\mathbf{a} \in \mathcal{Z}} \psi(\mathbf{a}). \tag{3}$$

3 Choice of the loss function

3.1 Common loss functions

Given the sample $\mathbf{Z}$, a naive but fast method to obtain an optimal clustering would be to consider the single partition that obtained the highest posterior value during the

sampling, i.e.:

$$\hat{\mathbf{a}}_{MAP} = \underset{t=1,2,\dots,T}{\arg\max} \, \pi \left(\mathbf{z}^{(t)} \big| \mathcal{Y} \right). \tag{4}$$

In a decision theoretic context, this is equivalent to choosing a $0-1$ loss defined as:

$$\mathcal{L}\left(\mathbf{a}, \mathbf{z}\right) = \begin{cases} 1 & \text{if } \mathbf{a} \not\equiv \mathbf{z}, \\ 0 & \text{if } \mathbf{a} \equiv \mathbf{z}; \end{cases} \tag{5}$$

since the Bayes action minimising (3) would simply be the mode of the sample. The sign "$\equiv$" here means that there exists a label permutation σ such that $\sigma\left(a_i\right) = z_i$, $\forall i \in \mathcal{V}$. Reading the definition in (5), the loss is zero iff the partitions are equivalent. In all of the other cases, the loss is 1 regardless of how different the partitions actually are. This peculiar behaviour makes the $0-1$ loss rather unappealing as means to compare partitions.

Note that all of the clustering algorithms that return a MAP estimate can be interpreted in this context as tools minimising the expected $0-1$ loss, although they normally do not require the sample $\mathbf{Z}$, and are computationally cheap. Hence MAP estimates may be criticised since in the Bayesian paradigm the corresponding loss is not particularly sensible.

Another loss function that is commonly used is the quadratic loss, which gives the posterior mean as Bayes action. However, in a clustering context this has little meaning due to the categorical nature of the variables, which makes any sort of averaging of allocations not particularly meaningful.

3.2   Loss functions to compare partitions

A more sensible approach would be to choose a loss function that is specifically designed to compare partitions. In recent years, many measures to compare partitions have been proposed, each with very different properties and characteristics. The works of Meilă (2007), Vinh et al. (2010), and Wade and Ghahramani (2015) and references therein offer an excellent overview.

A common approach used to compare partitions (here $\mathbf{a}$ and $\mathbf{z}$ denote two arbitrary partitions with $K_{\mathbf{a}}$ and $K_{\mathbf{z}}$ groups, respectively) relies on the $K_{\mathbf{a}} \times K_{\mathbf{z}}$ contingency matrix (or confusion matrix), whose entries are defined as:

$$n_{gh}^{\mathbf{a},\mathbf{z}} = \sum_{i=1}^{N} \mathbb{1}_{\{a_i=g\}} \mathbb{1}_{\{z_i=h\}} \tag{6}$$

where g varies among the groups of $\mathbf{a}$, and h among those of $\mathbf{z}$. The entries of such a matrix simply count the number of items that $\mathbf{a}$ classifies in group g and $\mathbf{z}$ classifies in group h, for every g and h.

Here, we focus on loss functions that depend on $\mathbf{a}$ and $\mathbf{z}$ only through the entries of $\mathbf{n}^{\mathbf{a,z}}$. This is a fairly general and reasonable assumption which is in line with the theory developed by Binder (1978); in fact, most metrics can be transformed into functions of the counts (see Vinh et al. (2009) and references therein).

We assume that the loss function has the following representation:

$$\mathcal{L}\left(\left\{n_{gh}^{\mathbf{a,z}}\right\}_{g,h}, \left\{n_g^{\mathbf{a}}\right\}_g, \left\{n_h^{\mathbf{z}}\right\}_h\right) = f_0\left(\sum_{g=1}^{K_{\mathbf{a}}}\sum_{h=1}^{K_{\mathbf{z}}} f_1\left(n_{gh}^{\mathbf{a,z}}\right), \sum_{g=1}^{K_{\mathbf{a}}} f_2\left(n_g^{\mathbf{a}}\right), \sum_{h=1}^{K_{\mathbf{z}}} f_3\left(n_h^{\mathbf{z}}\right)\right) \quad (7)$$

where f_0, f_1, f_2, f_3 are real valued functions that can be evaluated in constant time and $n_g^{\mathbf{a}}$ and $n_h^{\mathbf{z}}$ indicate the sizes of group g and h, respectively, i.e.:

$$n_g^{\mathbf{a}} = \sum_{h=1}^{K_{\mathbf{z}}} n_{gh}^{\mathbf{a,z}}, \qquad n_h^{\mathbf{z}} = \sum_{g=1}^{K_{\mathbf{a}}} n_{gh}^{\mathbf{a,z}}. \quad (8)$$

for every $g = 1, \ldots, K_{\mathbf{a}}$ and $h = 1, \ldots, K_{\mathbf{z}}$. The assumption determined by (7) is actually not restrictive: most of the commonly used loss functions for partitions satisfy this condition. We note that the arguments of the function f_0 include the following quantities as special cases:

- The entropies of $\mathbf{a}$ and $\mathbf{z}$, describing the uncertainty associated to $\mathbf{a}$ and $\mathbf{z}$, respectively:

$$H\left(\mathbf{a}\right) = -\sum_{g=1}^{K_{\mathbf{a}}} \frac{n_g^{\mathbf{a}}}{N} \log_2 \frac{n_g^{\mathbf{a}}}{N}; \qquad H\left(\mathbf{z}\right) = -\sum_{h=1}^{K_{\mathbf{z}}} \frac{n_h^{\mathbf{z}}}{N} \log_2 \frac{n_h^{\mathbf{z}}}{N}. \quad (9)$$

- The joint entropy of $\mathbf{a}$ and $\mathbf{z}$:

$$H\left(\mathbf{a}, \mathbf{z}\right) = -\sum_{g=1}^{K_{\mathbf{a}}}\sum_{h=1}^{K_{\mathbf{z}}} \frac{n_{gh}^{\mathbf{a,z}}}{N} \log_2 \frac{n_{gh}^{\mathbf{a,z}}}{N}. \quad (10)$$

  This describes instead the uncertainty of the random variable with pdf given by the quantities $n_{gh}^{\mathbf{a,z}}/N$, for every g and h.

- The mutual information, which can be evaluated from the entropies and joint entropy:

$$I\left(\mathbf{a}, \mathbf{z}\right) = H\left(\mathbf{a}\right) + H\left(\mathbf{z}\right) - H\left(\mathbf{a}, \mathbf{z}\right). \quad (11)$$

  This quantity is particularly meaningful and has been advocated in a normalised version by Strehl and Ghosh (2003) as a distance measure between partitions.

Note the common convention that $x \log_2 x = 0$ if $x = 0$. Evidently these information-based quantities can be obtained as special cases of the functions f_1, f_2 and f_3, making our assumption rather general and broadly satisfied.

Here follows a brief description of some well-known loss functions that can be considered with our approach.

Binder's loss (B). We use a special case of a more general formula first introduced by Binder (1978):

$$\mathcal{L}_B\left(\mathbf{a}, \mathbf{z}\right) = \frac{1}{2} \sum_{g=1}^{K_\mathbf{a}} \left(n_g^\mathbf{a}\right)^2 + \frac{1}{2} \sum_{h=1}^{K_\mathbf{z}} \left(n_h^\mathbf{z}\right)^2 - \sum_{g=1}^{K_\mathbf{a}} \sum_{h=1}^{K_\mathbf{z}} \left(n_{gh}^{\mathbf{a},\mathbf{z}}\right)^2. \tag{12}$$

This loss is equivalent to the Hamming distance (Meilă 2012) and to the Rand index (Rand 1971). Binder's loss has an interesting property that simplifies greatly the minimisation of (3). One can in fact easily construct a so-called posterior similarity matrix of size $N \times N$, whose entries b_{ij} denote the estimated posterior probability of i and j being allocated to the same group, for every i and j in $\mathcal{V}$. Then, the Binder Bayes action satisfies:

$$\hat{\mathbf{a}}_B = \arg \min_{\mathbf{a} \in \mathcal{Z}} \sum_{i<j} \left[\mathbb{1}_{\{a_i = a_j\}} - b_{ij} \right] \tag{13}$$

where $\mathbb{1}_\mathcal{A}$ is equal to 1 if the event $\mathcal{A}$ is true or zero otherwise. This simplifies the minimisation problem since (13) depends on the sample only through the posterior similarity matrix, which can be effectively computed beforehand.

The variation of information (VI). This loss is one we particularly focus on in this paper, and is defined as:

$$\mathcal{L}_{VI}\left(\mathbf{a}, \mathbf{z}\right) = 2H\left(\mathbf{a}, \mathbf{z}\right) - H\left(\mathbf{a}\right) - H\left(\mathbf{z}\right). \tag{14}$$

The VI loss, first studied in Meilă (2007), has received an increasing amount of attention in the last decade, mainly due to its strong mathematical foundations and practical efficiency. In the paper by Meilă (2007) as well as in subsequent works such as Wade and Ghahramani (2015), the mathematical properties and behaviour of the VI loss have been deeply studied. We mention that this loss is a metric, that it forms a lattice and that it is horizontally and vertically aligned in the space of partitions. In addition, it is invariant to label-switching, i.e. switching labels for either $\mathbf{a}$ or $\mathbf{z}$ will not affect the value $\mathcal{L}_{VI}\left(\mathbf{a}, \mathbf{z}\right)$. More details regarding the theoretical properties of the VI loss can be found in Meilă (2007).

The normalised variation of information (NVI). This loss is defined as:

$$\mathcal{L}_{NVI}\left(\mathbf{a}, \mathbf{z}\right) = 1 - \frac{I\left(\mathbf{a}, \mathbf{z}\right)}{H\left(\mathbf{a}, \mathbf{z}\right)}. \tag{15}$$

The normalised version of the VI loss takes values in $[0, 1]$. This scale-invariance may facilitate the interpretation and the comparisons of partitions under different conditions. Since we adopt an optimisation approach, this feature is not crucial in our framework due to the partitions always referring to the same set of individuals.

The normalised information distance (NID). This loss is defined as:

$$\mathcal{L}_{NID}\left(\mathbf{a}, \mathbf{z}\right) = 1 - \frac{I\left(\mathbf{a}, \mathbf{z}\right)}{\max\left\{H\left(\mathbf{a}\right), H\left(\mathbf{z}\right)\right\}}. \tag{16}$$

The NID loss has been advocated in Vinh et al. (2010) as a general purpose - context independent - loss function with desirable behaviours.

4 Minimisation of the expected posterior loss

An exhaustive search within $\mathcal{Z}$ becomes impractical even for very small N (the cardinality of $\mathcal{Z}$ is a number with more than 100 digits if $N = 100$). Therefore, the minimisation can be seen as a binary programming optimisation problem which is known to be NP-hard, and hence not solvable through exact methods.

Also, the objective function requires the calculation of the sum in (2) at each evaluation. Getting a new posterior sample at each step is not a practical option, hence the same sample is used for all of the evaluations of (2). Nonetheless, even a single evaluation of the objective function can become computationally burdensome when the size of the sample is large. Therefore, the decision theoretic approach becomes soon impractical as N and T increase, and finding scalable procedures is crucial. In this section we introduce a new algorithm that, using greedy updates, is able to estimate the Bayes action for the wide family of loss functions satisfying (7), requiring in input only the posterior sample of partitions.

4.1 Greedy algorithm

Heuristic greedy algorithms have been recently rediscovered as a means to maximise the so-called *exact Integrated Complete Likelihood* in various contexts: stochastic block models (Côme and Latouche 2015), latent block models (Wyse et al. 2014), Gaussian finite mixtures (Bertoletti et al. 2015). Similar approaches have also been proposed in Bayesian nonparametrics for Dirichlet prior mixtures (Raykov et al. 2014) although in this case they did not cast the clustering problem into the optimisation of an exact model-based clustering criterion. Among the many papers adopting types of greedy optimisation, we find the approaches of Besag (1986), Strehl and Ghosh (2003), and Newman (2004) particularly related to ours.

We propose a greedy algorithm that updates a partition by changing the cluster memberships of single observations using a greedy heuristic, hence decreasing the expected posterior loss of the partition at each step. As input, the algorithm only requires a starting partition, the posterior sample $\mathbf{Z}$ and a user-specified parameter K_{up}, equal to the maximum number of groups allowed (a reasonable default value would be $K_{up} = N$). The algorithm cycles over the observations in random order, and, for each of these, it tries all

of the possible reallocations, eventually choosing the one giving the best decrease in the objective function. The notation $\mathbf{a}_{i:r\to s}$ denotes the partition $\mathbf{a}$ where the observation i has been reallocated from group r to s. At each move, the number of groups may increase (if the observation is reallocated to an empty group) or decrease (if a group is left empty), although the latter scenario is much more frequent. Due to the low probability of creating new groups, it is generally advisable to start with a partition made of close to K_{up} groups. The procedure stops when a complete sweep over all observations yields no change in the expected posterior loss. The pseudo-code for the algorithm is shown in Algorithm 1.

Algorithm 1 Greedy algorithm

1: Let $\mathbf{a}$ be the starting partition.
2: Set $\psi_{\mathbf{a}} = \psi(\mathbf{a})$.
3: Set STOP to $false$.
4: **while** STOP is $false$ **do**
5:     $\psi_{stop} = \psi_{\mathbf{a}}$.
6:     Set $\mathcal{V} = \{1, 2, \ldots, N\}$.
7:     **while** $\mathcal{V}$ is not empty **do**
8:        Pick i at random from $\mathcal{V}$ and delete it from $\mathcal{V}$.
9:        For every $s = 1, \ldots, K_{up}$, evaluate $\psi(\mathbf{a}_{i:a_i\to s})$.
10:        Move i to $\hat{s} = \underset{s=1,\ldots,K_{up}}{\arg\max}\, \psi(\mathbf{a}_{i:a_i\to s})$.
11:        Update $\psi_{\mathbf{a}} = \psi(\mathbf{a})$.
12:     **end while**
13:     **if** $\psi_{stop} = \psi_{\mathbf{a}}$ **then**
14:        STOP $= true$.
15:     **end if**
16: **end while**
17: Return $\mathbf{a}$ and $\psi_{\mathbf{a}}$.

Due to the greedy nature of this procedure, the algorithm is bound to return a local optimum, rather than a global one. Consequently, several restarts with different initial partitions may be required. However, convergence is usually reached in very few iterations, in each run. Regarding the starting partition, this may either be chosen at random or it may be set to be the clustering yielding the highest posterior value as in (4). A possible alternative lies in between the two cases, i.e. the MAP partition may be changed to some extent by reallocating some observations at random.

One interesting feature of the greedy algorithm is that the whole space of partitions is explored, hence the optimal partitions may differ substantially from all of the clusterings in the sample. In fact, many non-optimal solutions may have higher posterior values than

the optimal one. In contrast to Côme and Latouche (2015) and Wyse et al. (2014), we do not perform any final merge step, as in most cases this did not improve the results.

4.2 Complexity

The basic operation that determines the complexity of the greedy optimisation is the evaluation of the variation in the objective function when a possible reallocation is tested (line 9 in the pseudo-code 1). Assume that the move from $\mathbf{a}$ to $\mathbf{a}_{i:r \to s}$ is being tested, for some groups r and s. The following quantity needs to be evaluated:

$$\Delta \psi := \psi\left(\mathbf{a}_{i:r \to s}\right) - \psi\left(\mathbf{a}\right) = \frac{1}{T} \sum_{t=1}^{T} \left[\mathcal{L}\left(\mathbf{a}_{i:r \to s}, \mathbf{z}^{(t)}\right) - \mathcal{L}\left(\mathbf{a}, \mathbf{z}^{(t)}\right)\right], \qquad (17)$$

which in turn requires, $\forall t = 1, \ldots, T$:

$$\Delta \mathcal{L}^{(t)} := \mathcal{L}\left(\mathbf{a}_{i:r \to s}, \mathbf{z}^{(t)}\right) - \mathcal{L}\left(\mathbf{a}, \mathbf{z}^{(t)}\right). \qquad (18)$$

For a certain t, the move only affects two entries of $\mathbf{n^a}$ (i.e. $n_r^{\mathbf{a}}$ and $n_s^{\mathbf{a}}$) and two entries of $\mathbf{n^{a,z}}^{(t)}$ (i.e. $n_{rv}^{\mathbf{a,z}^{(t)}}$ and $n_{sv}^{\mathbf{a,z}^{(t)}}$, where $v = z_{ti}$). This means that the change in the arguments of f_0 can be evaluated in a constant time, hence making the cost of evaluating $\Delta \psi \sim \mathcal{O}\left(T\right)$.

Since the algorithm tries all possible moves for each observation, the overall computational cost is $\mathcal{O}\left(TNK_{up}\right)$.

4.3 Comparisons with other algorithms

Both Lau and Green (2007) and Wade and Ghahramani (2015) propose original algorithmic frameworks to minimise an expected posterior loss. While Lau and Green (2007) only focus on Binder's loss, Wade and Ghahramani (2015) also extend the procedure to the VI loss, albeit resorting to an approximation of the objective function. Both methodologies take advantage of the posterior similarity matrix representation, briefly pointed out in (13). Note that this representation is exclusive to the Binder's loss, hence these approaches lack the possibility to be generalised to other loss functions, unless approximations are introduced.

The computational cost for an evaluation of the objective function (13) does not depend on T, since the posterior information contained in the sample is summarised in the posterior similarity matrix. The calculation of the posterior similarity matrix itself requires $\mathcal{O}\left(TN^2\right)$ operations, yet this can be performed offline and it is unlikely to impact the overall computing time.

On the other hand, our algorithm does not require a N^2 cost at any stage, hence it should be preferrable when the number of observations to classify is very large. We note that, due to the dependence of the complexity on T, our algorithm will benefit if

the sample is small and thinned with a large lag. A trade-off between the reliability of the posterior sample and computing time should be assessed, in that one should provide a sample that is as small as possible but not so small that the approximation to the posterior distribution is not reliable. As concerns the dependency on K_{up}, ideally one should choose $K_{up} = N$, but this evidently would make the procedure impractical in large N scenarios.

More generally, the computational cost of the algorithm may be compared to the complexity of the sampler used to get the posterior sample. In fact, one key advantage of the collapsed Gibbs samplers proposed in Nobile and Fearnside (2007), McDaid et al. (2013), and Wyse and Friel (2012) is their computational efficiency. The posterior sample returned by these samplers is necessary to perform the minimisation of the expected posterior loss. Hence, an ideal complexity for the optimisation problem should be not higher than that required by the sampler in the first place. Unfortunately, when analysing these samplers, new quantities (the number of dimensions for Gaussian finite mixtures, or the number of edges in block models) come into play, making a strict comparison of the complexity not possible. However, in our applications we noticed that for stochastic block models and latent block models the computational bottleneck was set by the samplers, and not by the greedy algorithm.

5   Classes of equivalences in the posterior sample

Since the sample space $\mathcal{Z}$ is discrete, the posterior sample $\mathbf{Z}$ may contain repetitions, due to the sampler returning to the same partition during the sampling procedure. This suggests that, regardless of the partition $\mathbf{a}$, a number of the calculations required to obtain $\mathcal{L}(\mathbf{a}, \mathbf{z})$ is redundant. In fact, given a partition $\mathbf{z}$, the following holds:

$$\mathcal{L}\left(\mathbf{a}, \mathbf{z}^{(t)}\right) = \mathcal{L}\left(\mathbf{a}, \mathbf{z}\right); \tag{19}$$

$$\mathcal{L}\left(\mathbf{a}_{i \to g}, \mathbf{z}^{(t)}\right) = \mathcal{L}\left(\mathbf{a}_{i \to g}, \mathbf{z}\right). \tag{20}$$

for all $i = 1, \ldots, N$ and $g = 1, \ldots, K_{up}$ and $\forall t : \mathbf{z}^{(t)} \equiv \mathbf{z}$.

It follows that the posterior sample can be summarised into the sample of its unique rows $\tilde{\mathbf{Z}} = \left\{ \tilde{\mathbf{z}}^{(1)}, \ldots, \tilde{\mathbf{z}}^{(\tilde{T})} \right\}$ and a vector of counts $\boldsymbol{\omega} = \left\{ \omega^{(1)}, \ldots, \omega^{(\tilde{T})} \right\}$ describing how many times the corresponding partition appears in the original sample $\mathbf{Z}$. Therefore the approximate expected posterior loss can be equivalently written as:

$$\psi\left(\mathbf{a}\right) = \frac{1}{\tilde{T}} \sum_{t=1}^{\tilde{T}} \omega^{(t)} \mathcal{L}\left(\mathbf{a}, \tilde{\mathbf{z}}^{(t)}\right). \tag{21}$$

A similar reasoning can be used to make the calculation of $\psi\left(\mathbf{a}_{i \to g}\right)$ more efficient.

The main difficulty in applying the technique just described lies in identifying the new representation efficiently. One problem consists in the implementation of the operator

"≡" since partitions should be compared up to a permutation of the labels. To solve this, we use a procedure described in Strehl and Ghosh (2003) that defines a unique labelling for all partitions: the first item is assigned to cluster 1, and then iteratively the next item is assigned either to an existing cluster or to the next empty cluster. Using this re-labelling, any two equivalent partitions will be transformed into the same sequence of digits in a computational time $\mathcal{O}(TN)$.

Furthermore, the same vector can be seen as a number in base$-K_{up}$ representation which uniquely identifies the corresponding partition and the equivalence class imposed by "≡". Hence a sorting algorithm can be used to reorder the sample according to such identifiers, for a computational cost of $\mathcal{O}(NT\log T)$, where N is the cost of a single comparison of partitions. Once the partitions are sorted, the unique set and the corresponding weights can be obtained in $\mathcal{O}(TN)$.

The advantage provided by this representation heavily depends on the dataset and on the corresponding marginal posterior distribution: less repetitions will appear if the posterior is flat and the partitioning very uncertain. On the other hand, the computational savings may be substantial in cases where only few partitions have a high posterior value.

Note that the sorting procedure creates a new computational bottleneck in the case where $\log T > K_{up}$. However, we found this is not relevant in practical terms and negligible when compared to the computational time demanded by the actual optimisation.

In a machine learning context, the weighted sample $\tilde{\mathbf{Z}}$ may be interpreted as a cluster ensemble problem, whereby each partition corresponds to the output of a clustering algorithm and the counts are weights describing the relative (possibly subjective) importance of the solution. Our methodology may be applied in this scenario without further modifications, providing a sound background to the decision making process.

6   Real data examples

In this Section, we provide three applications of our methodology to different clustering contexts, and compare the results obtained with previous analyses. To avoid confusion, we show the results only for the VI loss, and note that the other losses lead to similar partitions.

6.1   Galaxies' dataset

6.1.1   The data

The dataset considered is composed of the velocities of 82 distant galaxies diverging from the Milky Way. Interest lies in understanding whether velocity can be used to discriminate clusters of galaxies. The dataset has been first analysed from a statistical point of view in Roeder (1990), and has been re-proposed in numerous papers dealing

with mixture models, including Richardson and Green (1997), Stephens (2000), and Wade and Ghahramani (2015).

6.1.2 The model

The observed data is denoted by $\mathcal{Y} = \{y_1, \ldots, y_N\}$, where $N = 82$ and $y_i \in \mathbb{R}$ for every $i = 1, \ldots, N$. As in Bertoletti et al. (2015), a Gaussian finite mixture model is adopted:

$$p\left(\mathcal{Y}|\boldsymbol{\lambda}, \boldsymbol{\mu}, \mathbf{r}\right) = \prod_{i=1}^{N} \sum_{g=1}^{K} \lambda_g \mathcal{N}\left(y_i; \ \mu_g, \ \frac{1}{r_g}\right); \tag{22}$$

where $\lambda_1, \ldots, \lambda_g$ are the mixture weights and $\mathcal{N}\left(\ \cdot\ ; \mu, \frac{1}{r}\right)$ denotes the univariate Gaussian distribution with mean μ and variance $1/r$. The number K of Gaussian components in the mixture is unknown and hence to be inferred.

Following a latent variable framework, an allocation variable z_i is associated to each observation, denoting which Gaussian component has generated the corresponding y_i:

$$p\left(y_i|z_i = g, \boldsymbol{\mu}, \mathbf{r}\right) = \mathcal{N}\left(y_i; \ \mu_g, \ \frac{1}{r_g}\right). \tag{23}$$

This allows a more tractable expression for the likelihood, conditionally on the allocations:

$$p\left(\mathcal{Y}|\mathbf{z}, \boldsymbol{\mu}, \mathbf{r}\right) = \prod_{g=1}^{K} \prod_{i:z_i=g} \mathcal{N}\left(y_i; \ \mu_g, \ \frac{1}{r_g}\right). \tag{24}$$

We specify a Bayesian hierarchical structure on both the likelihood parameters and the allocation variables. For every group g, the parameters r_g and μ_g are independent realisations of a $\text{Gamma}(\gamma, \delta)$ and a $\text{Gaussian}\left(0, [\tau r_g]^{-1}\right)$, respectively. As concerns the allocations, these are distributed as independent Multinomials with parameters $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_K)$, where $\boldsymbol{\theta}$ is a Dirichlet distributed random vector. The hyperparameters are set as in Bertoletti et al. (2015): $\tau = 0.01$, $\gamma = 0.5$, $\delta = 0.5$, and Dirichlet hyperparameter $\alpha = 4$.

Since conjugate priors are used, most of the model parameters can be integrated out analytically. Hence the following marginal distributions can be obtained in exact form for the data and allocations:

$$p\left(\mathcal{Y}|\mathbf{z}, \tau, \gamma, \delta\right) = \prod_{g=1}^{K} \int_0^{\infty} p\left(r_g|\gamma, \delta\right) \int_{-\infty}^{+\infty} p\left(\mu_g|\tau, r_g\right) \prod_{i:z_i=g} p\left(y_i|z_i = g, \mu_g, r_g\right) d\mu_g dr_g; \tag{25}$$

$$p\left(\mathbf{z}|\alpha\right) = \int_{\Theta} p\left(\mathbf{z}|\boldsymbol{\theta}\right) p\left(\boldsymbol{\theta}|\alpha\right) d\boldsymbol{\theta}. \tag{26}$$

More details on the integrations can be found in Nobile and Fearnside (2007) and Bertoletti et al. (2015). A consequence of these results is that the marginal posterior for the allocations can be obtained analytically, too:

$$p\left(\mathbf{z}|\mathcal{Y}\right) \propto p\left(\mathcal{Y}|\mathbf{z}, \tau, \gamma, \delta\right) p\left(\mathbf{z}|\alpha\right) \tag{27}$$

Such marginal posterior distribution can be used as target in a Markov chain Monte Carlo sampler, thereby obtaining the posterior sample $\mathbf{Z}$. Note that, since all of the model parameters have been integrated out, trans-dimensional moves can be easily implemented, so that the chain effectively explores all of the possible models. In Appendix A.1, a general algorithm to sample from this distribution is described. The same sampler is used to get the posterior sample $\mathbf{Z}$ for the galaxies' dataset.

6.1.3 Results

We obtained a sample for the allocations using the collapsed sampler described in Appendix A.1. One million observations were first discarded as burn-in, then one observation every hundredth was retained until a sample size of 10,000 was obtained. The chain appeared to mix well suggesting convergence to the target distribution. The sample was post-processed using the method described in Section 5. Then, several runs of the greedy algorithm were performed, using both the noisy MAP and completely random starting partitions. The left panel of Figure 1 shows a histogram of the observed data with the overall best clustering found. The number of groups for the VI Bayes action is 3, which
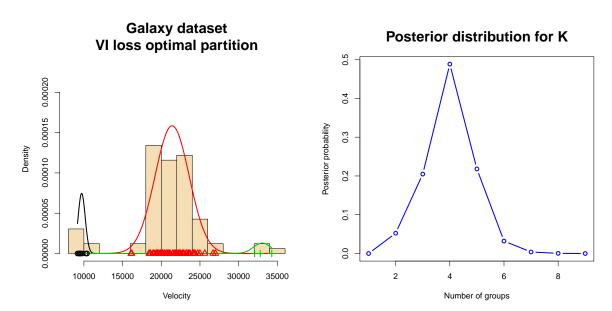

Figure 1: **Galaxy dataset**. *On the left panel, the VI loss best partition found is shown. The right panel shows the posterior probabilities for the number of groups. The distribution has a peak at $K = 4$, which contrasts with the number of clusters in the optimal partition, equal to 3.*

is in line with the results of Wade and Ghahramani (2015), but notably different from the results of Richardson and Green (1997).

The computational time needed to get the sample was about 45 seconds, whereas an average of 5 seconds was required for each run of the greedy algorithm, with K_{up} fixed to 50 for both algorithms.

6.2 Stochastic block models: French political blogosphere

6.2.1 The data

The data first appeared in Zanghi et al. (2008) and consist of a undirected graph where nodes represent political blogs' websites and edges represent hyperlinks between them. As in Latouche et al. (2011) we focus only on a subset of the original dataset, available in the R package `mixer`. The data consist of a single day snapshot of political blogs automatically extracted on the 14th of October 2006 and manually classified by the "Observatoire Présidentielle project". The graph is composed of 196 nodes and 1432 edges, and the main political parties are the UMP (French "republican"), UDF ("moderate" party), liberal party (supporters of economic liberalism) and PS (French "democrat"), although 11 different parties appear in total. The observed data is modelled by the adjacency matrix $\mathcal{Y}$ whose entries are defined as follows:

$$
y_{ij} = \begin{cases} 1 & \text{if an undirected edge between blogs } i \text{ and } j \text{ appear;} \\ 0 & \text{otherwise;} \end{cases} \tag{28}
$$

for every $1 \le i < j \le N$.

6.2.2 Stochastic block models

Stochastic block models (Nowicki and Snijders 2001) are finite mixture models for networks, whereby the clustering problem is formulated on the nodes of the network and the connection profile of each node is selected by its cluster membership. For every i, the allocation variable z_i denotes the group to which node i belongs, and, as in the Gaussian finite mixture context, a Multinomial-Dirichlet structure is assumed on these variables. The number of underlying groups K is unknown and hence to be inferred. Conditionally on the allocations, the likelihood for the graph $\mathcal{Y} = \{y_{ij} : 1 \le i < j \le N\}$ factorises as:

$$
P\left(\mathcal{Y}|\mathbf{z}, \Pi\right) = \prod_{g=1}^{K}\prod_{h=1}^{K} \prod_{\{i:z_i=g\}} \prod_{\substack{\{j:z_j=h\} \\ j \ne i}} \pi_{gh}^{y_{ij}} \left(1 - \pi_{gh}\right)^{1-y_{ij}}. \tag{29}
$$

Here, Π is a symmetric $K \times K$ matrix of connection probabilities, where the generic element π_{gh} indicates the probability that an edge occurs between a node in group g and a node in group h, for any g and h in $\{1, \ldots, K\}$. Furthermore, each π_{gh} is assumed to be a realisation of an independent Beta random variable. The hyperparameters for the Beta and Dirichlet distributions are all set to 0.5.

Since conjugate priors are used, all of the model parameters can be integrated out analytically. It follows that, as in the Gaussian finite mixture context, the quantity $p\left(\mathbf{z}|\mathcal{Y}\right)$ is available anaylitically and can be targeted by the sampler described in Appendix A.1. Further details on the integration can be found in McDaid et al. (2013) and Côme and Latouche (2015).

6.2.3 Results

First, we performed block modelling using the variational algorithm implemented in the package `mixer` and obtained a partitioning to be used as reference. The optimal variational solution has 12 groups, which roughly correspond to the political affiliations, as shown in Table 1.

Table 1: French blogs: confusion matrix for the variational partition and the political affiliations.

	1	2	3	4	5	6	7	8	9	10	11	12
Cap21	2	0	0	0	0	0	0	0	0	0	0	0
CA	0	0	8	0	0	0	0	0	0	0	0	3
FN - MNR - MPF	4	0	0	0	0	0	0	0	0	0	0	0
Les Verts	5	0	2	0	0	0	0	0	0	0	0	0
PCF - LCR	5	0	1	0	0	0	0	0	0	0	0	0
PCF LCR	1	0	0	0	0	0	0	0	0	0	0	0
PS	5	0	9	0	0	0	19	18	2	4	0	0
PRG	9	0	1	0	0	0	0	1	0	0	0	0
UDF	0	1	1	0	24	6	0	0	0	0	0	0
UMP	1	24	2	11	2	0	0	0	0	0	0	0
liberaux	1	0	0	0	0	0	0	0	0	0	24	0

Then, we used our methodology to estimate the VI loss optimal partition. The sample for the allocation variables was obtained through the Collapsed SBM algorithm of McDaid et al. (2013), discarding the first 1 million updates and keeping 1 observation every 100th thereafter. A sample size of 10,000 was then used to perform the greedy optimisation, using both noisy MAP and random starting partitions. The computational time needed to get the sample was about 5 hours, whereas an average of 50 seconds was required for each run of the greedy algorithm, with K_{up} fixed to 50.

The VI-optimal partition exhibits 18 groups, and is represented in the right panel of Figure 2. Figure 3 shows instead the reordered adjacency matrices for the three different partitions. The posterior distribution for the number of groups is shown in Figure 4. As in the galaxies' dataset, the optimal number of group contrasts with the modal value of the posterior distribution.

It appears that the VI-optimal clustering is a finer partition that splits up some of the larger groups into subgroups. Nonetheless from Figure 3 it is clear that this entails a better discrimination of the profiles of blogs. A confusion matrix matching the solution to the political affiliations is shown in Table 2. The liberals are well discriminated in both the variational and VI-optimal partitions. The two partitions also agree on the blogs affiliated to the UDF party: 24 of them are well-discriminated and isolated from the rest, a subset of 6 blogs are classified into their own group, 1 blog is associated to the UMP party and 1 is not well-recognised. The main differences between the two partitions arise with respect to the other two relevant parties: UMP and PS. In these two cases it appears that the relational profiles of the blogs are not particularly determined by the political
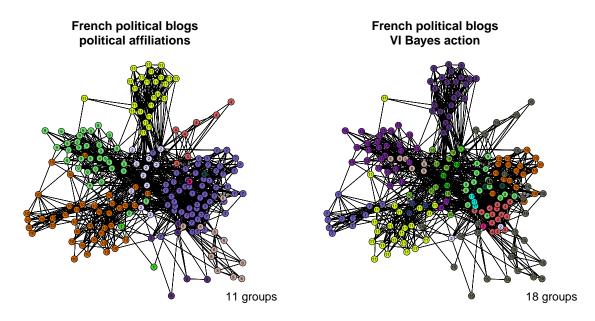
French political blogs
political affiliations

11 groups

French political blogs
VI Bayes action

18 groups

Figure 2: **French blogs.** *Representation of the French political blogs network, with colours and node labels denoting cluster memberships.*

affiliation, since both partitions recognise a number of subgroups within each party, signaling heterogeneity. UMP is decomposed in 5 subgroups in both partitions, while PS is decomposed in 6 and 7 subgroups for the variational and VI partition, respectively.

6.3   Latent block model: Congressional voting data

We propose an application of our methodology to the UCI Congressional voting data, previously analysed in Wyse and Friel (2012) and Wyse et al. (2014).
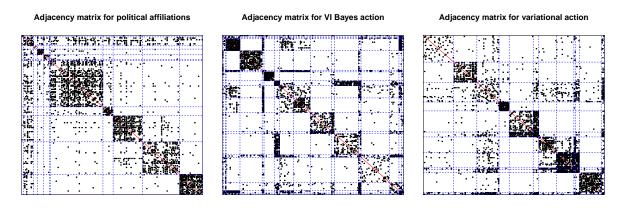

Adjacency matrix for political affiliations   Adjacency matrix for VI Bayes action   Adjacency matrix for variational action

Figure 3: **French blogs.** *Reordered adjacency matrices for three different partitioning of the French political blogs dataset: available political affiliations (left panel), VI-optimal allocations (central panel) and variational optimal allocations (right panel).*
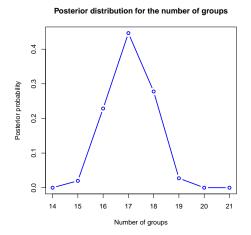
17

Figure 4: **French blogs.** *Posterior distribution for the number of groups in the French political blogosphere dataset. The MAP value is $K = 17$ which contrasts with the optimal value obtained through the greedy algorithm.*

Table 2: **French blogs.** *Confusion matrix for the VI-optimal partition and the political affiliations.*

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Cap21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0
CA	1	0	0	0	0	0	0	0	0	0	0	0	0	3	0	6	0	1
FN-MNR-MPF	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0
Les Verts	0	0	0	0	0	0	0	0	2	0	0	0	0	0	5	0	0	0
PCF-LCR	0	1	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0
PCF LCR	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
PS	0	0	0	15	0	2	0	0	13	18	0	0	0	0	5	1	3	0
PRG	0	1	1	0	0	0	0	0	0	0	0	0	0	0	9	0	0	0
UDF	0	0	0	0	0	0	0	6	0	0	1	0	24	0	0	0	0	1
UMP	0	0	0	0	0	0	11	0	0	0	21	3	2	0	0	3	0	0
liberaux	0	0	0	0	24	0	0	0	0	0	1	0	0	0	0	0	0	0

6.3.1 The data

The data record whether 435 members of the 98th congress voted "yay" or "nay" on 16 key issues. Abstained and absent were treated as "nays". Also, information on the political affiliation of each member is available: 267 individuals are "democrats" and 168 "republicans". Following Wyse and Friel (2012) and Wyse et al. (2014), the data are rearranged into a bipartite network, whereby two types of nodes are defined (one corresponding to congress members and one to issues) and only undirected edges between nodes of different types are allowed. Similarly to stochastic block models, an adjacency matrix $\mathcal{Y}$ is used to summarise the data, with edges corresponding to "yays" ($y_{ij} = 1$) and non-edges corresponding to "nays" ($y_{ij} = 0$). Note that in this case the matrix $\mathcal{Y}$ has size 435×16, whereby rows correspond to congressmen and columns to issues.

6.3.2 Bipartite latent block model

A latent block model (see, for instance, Wyse et al. (2014)) is used to model the bipartite graph. A clustering problem is formulated on both the rows and columns of the adjacency matrix: two partitions $\mathbf{r}$ and $\mathbf{c}$ determine the clustering of congress members and issues, respectively. The number of groups of $\mathbf{r}$ and $\mathbf{c}$ are denoted by K_r and K_c, respectively, and are unknown. These two partitions independently follow the same Multinomial-Dirichlet structure as described in previous applications.

As concerns the likelihood of the model, a $K_r \times K_c$ matrix Π is introduced, so that its generic element $\pi_{gh} \in [0, 1]$ corresponds to the probability of the occurance of an edge from a node in group g to a node in group h. Hence, conditionally on the allocations, the likelihood can be factorised into independent blocks:

$$P\left(\mathcal{Y}|\mathbf{r}, \mathbf{c}, \Pi\right) = \prod_{g=1}^{K_r}\prod_{h=1}^{K_c} \prod_{\{i:r_i=g\}} \prod_{\{j:c_j=h\}} \pi_{gh}^{y_{ij}}\left(1 - \pi_{gh}\right)^{1-y_{ij}}. \tag{30}$$

Bipartite latent block models may also be recast as finite mixture models, where the mixture is with respect to the partitions:

$$P\left(\mathcal{Y}|\boldsymbol{\theta}, \Pi\right) = \sum_{\mathbf{r},\mathbf{c}} p\left(\mathbf{r}|\boldsymbol{\theta}\right) p\left(\mathbf{c}|\boldsymbol{\theta}\right) P\left(\mathcal{Y}|\mathbf{r}, \mathbf{c}, \Pi\right). \tag{31}$$

The connection probabilities π_{gh} are realisations of independent Beta random variables for every $g = 1, \ldots, K_r$ and $h = 1, \ldots, K_c$, and all of the hyperparameters are fixed to 0.5.

Since conjugate priors are used, all of the model parameters can be integrated out analytically, thereby obtaining the marginal posterior $p\left(\mathbf{r}, \mathbf{c}|\mathcal{Y}\right)$ in exact form. Further details on the integration can be found in Wyse and Friel (2012) and Wyse et al. (2014).

6.3.3 Results

The algorithm of Wyse and Friel (2012) was used to obtain a sample for the allocations of both congress members and issues. Similarly to previous analyses, 1 million observations were discarded and 10,000 were used as final sample using a thinning of 100. The partitioning of the data corresponding to the highest posterior value was saved as a reference. We found that posing a clustering problem on the issues was not particularly interesting in that very few issues were aggregated in the same cluster, hence we will here show only the cluster analysis on the congress members. The sample of partitions for the members was processed through the procedure of Section 5, and then several runs of the greedy optimisation were performed. The computational time needed to get the sample was about 30 hours, whereas an average of 70 seconds was required for each run of the greedy algorithm, with K_{up} fixed to 30. Figure 5 shows the posterior sample for the number of groups. The reordered adjacency matrices for the MAP and the VI-optimal partition
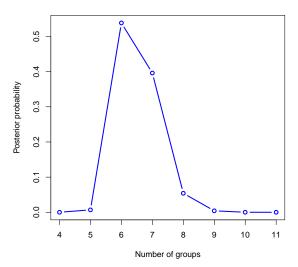
Posterior distribution for K

Figure 5: ***Congressional voting data****. Posterior distribution for the number of groups of congress members. The VI-optimal value $K = 6$ corresponds to the modal value, and the distribution is noticeably right-skewed.*

are shown in Figure 6. From the confusion table shown in Table 3 it appears that the two main political factions are split into 3 subgroups each, with a total of 29 individuals against the tide.

Table 3: ***Congressional voting data.*** *Confusion matrix comparing the political affiliation with the VI-optimal partition.*

	1	2	3	4	5	6
democrat	42	79	127	16	1	2
republican	4	6	0	125	30	3

7 Conclusions

We have proposed a Bayesian approach to summarise a sample of partitions from an arbitrary clustering context. We have described a greedy algorithm capable of finding the optimal partition in a wide range of clustering frameworks. The algorithm can handle many well-known loss functions. In our analyses, we focused on the variation of information loss, which has proven to be particularly effective in the optimisation context.

One appealing advantage of our methodology is that it can scale well with the number of items to be classified, hence being a useful general tool to use in an arbitrary clustering context. In fact, since previous methods focused only on particular choices of the loss function, our methodology is the only scalable method that can encompass most comparison measures within a unified framework. Also, label-switching issues do not affect
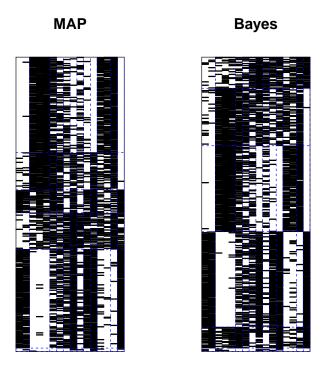
MAP   **Bayes**

Figure 6: **Congressional voting data.** *Reordered adjacency matrices for the MAP and the VI-optimal partitions. The partitions on the columns (issues) are equivalent, whereas the rows are clustered in different ways, although the number of clusters is equal.*

our method.

The greedy algorithm usually converges with very few iterations, however several restarts are useful to avoid convergence to local optima. We noticed that, when compared to other similar greedy routines (Côme and Latouche 2015; Wyse et al. 2014; Bertoletti et al. 2015), the algorithm is more likely to converge to the global optimum, even though no final hierarchical merge step is used. This may be a consequence of the fact that the objective function is generally smoother and easier to optimise.

The wide applicability of our algorithm comes at a cost: each step of the optimisation process involves a computational cost depending on the size of the sample T, which can easily make the problem intractable if a large sample is used. However, in most cases this impasse can be downsized simply by "thinning" the sample. As concerns storage costs, the the main bottleneck is set by the T contingency tables of size K^2 that are used throughout the optimisation.

To emphasise the context-independence of our approach, we have proposed applications to real datasets for three different clustering frameworks. In the Gaussian finite mixture case (galaxies' dataset), the results look interesting and in line with the work of Wade and Ghahramani (2015). The results on the French political blogosphere appear to be very different from those obtained through previous analyses. On one hand an overestimation of the number of groups may be argued, on the other the groups obtained with

our approach are evidently more homogeneous. A clustering problem on the members of the congressional voting data has also been proposed: here the two main political factions are well recognised and the results seem to agree with the previous analyses of Wyse and Friel (2012) and Wyse et al. (2014).

For each of the dataset analysed in this paper we have obtained the marginal sample for the allocation variables using a collapsed Gibbs sampler, which is a tool able to explore a number of models at the same time. This type of approach aims at improving the mixing of the Markov chain while keeping a low computational cost, and it generally works well in many clustering frameworks. However, due to the discrete nature of the sampled variables, rarely the sampler achieves good acceptance rates, and in some cases this causes a very slow mixing of the chain. This in turn biases the results obtained through the loss function approach, since our method heavily relies on the good quality of the sample of partitions. Unfortunately, at the moment there are no good solutions to address this impasse, suggesting that future research should focus on introducing new ways to explore the space of partitions $\mathcal{Z}$ in a clever way, hence making MCMC approaches more efficient.

Acknowledgements

References

Barry, D. and J. A. Hartigan (1992). "Product partition models for change point problems". In: *The Annals of Statistics*, pp. 260–279.

Benson, A. and N. Friel (2016). "An adaptive MCMC method for multiple changepoint analysis with applications to large datasets". In: *arXiv preprint arXiv:1606.09419*.

Bertoletti, M., N. Friel, and R. Rastelli (2015). "Choosing the number of clusters in a finite mixture model using an exact integrated completed likelihood criterion". In: *METRON*, pp. 1–23.

Besag, J. (1986). "On the statistical analysis of dirty pictures". In: *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 259–302.

Binder, D. A. (1978). "Bayesian cluster analysis". In: *Biometrika* 65.1, pp. 31–38.

Côme, E. and P. Latouche (2015). "Model selection and clustering in stochastic block models based on the exact integrated complete data likelihood". In: *Statistical Modelling*, p. 1471082X15577017.

Dahl, D. B. (2009). "Modal clustering in a class of product partition models". In: *Bayesian Analysis* 4.2, pp. 243–264.

Favaro, S. and Y. W. Teh (2013). "MCMC for normalized random measure mixture models". In: *Statistical Science* 28.3, pp. 335–359.

Friel, N., C. Ryan, and J. Wyse (2013). "Bayesian model selection for the latent position cluster model for Social Networks". In: *arXiv preprint arXiv:1308.4871*.

Fritsch, A. and K. Ickstadt (2009). "Improved criteria for clustering based on the posterior similarity matrix". In: *Bayesian analysis* 4.2, pp. 367–391.

Gionis, A., H. Mannila, and P. Tsaparas (2007). "Clustering aggregation". In: *ACM Transactions on Knowledge Discovery from Data (TKDD)* 1.1, p. 4.

Govaert, G. (1995). "Simultaneous clustering of rows and columns". In: *Control and Cybernetics* 24, pp. 437–458.

Green, P. J. (1995). "Reversible jump Markov chain Monte Carlo computation and Bayesian model determination". In: *Biometrika* 82.4, pp. 711–732.

Hartigan, J. A. (1990). "Partition models". In: *Communications in Statistics-Theory and Methods* 19.8, pp. 2745–2756.

Latouche, P., E. Birmelé, and C. Ambroise (2011). "Overlapping stochastic block models with application to the french political blogosphere". In: *The Annals of Applied Statistics*, pp. 309–336.

Lau, J. W. and P. J. Green (2007). "Bayesian model-based clustering procedures". In: *Journal of Computational and Graphical Statistics* 16.3, pp. 526–558.

MacDonald, I. L. and W. Zucchini (1997). *Hidden Markov and other models for discrete-valued time series*. Vol. 110. CRC Press.

McDaid, A. F., T. B. Murphy, N. Friel, and N. J. Hurley (2013). "Improved Bayesian inference for the stochastic block model with application to large networks". In: *Computational Statistics & Data Analysis* 60, pp. 12–31.

McLachlan, G. and D. Peel (2004). *Finite mixture models*. John Wiley & Sons.

Meilă, M. (2007). "Comparing clusterings: an information based distance". In: *Journal of multivariate analysis* 98.5, pp. 873–895.

Meilă, M. (2012). "Local equivalences of distances between clusterings: a geometric perspective". In: *Machine Learning* 86.3, pp. 369–389.

Neal, R. M. (2000). "Markov chain sampling methods for Dirichlet process mixture models". In: *Journal of computational and graphical statistics* 9.2, pp. 249–265.

Newman, M. E. J. (2004). "Fast algorithm for detecting community structure in networks". In: *Physical review E* 69.6, p. 066133.

Nobile, A. and A. T. Fearnside (2007). "Bayesian finite mixtures with an unknown number of components: The allocation sampler". In: *Statistics and Computing* 17.2, pp. 147–162.

Nowicki, K. and T. A. B. Snijders (2001). "Estimation and prediction for stochastic blockstructures". In: *Journal of the American Statistical Association* 96.455, pp. 1077–1087.

Quintana, F. A. (2006). "A predictive view of Bayesian clustering". In: *Journal of Statistical Planning and Inference* 136.8, pp. 2407–2429.

Rand, W. M. (1971). "Objective criteria for the evaluation of clustering methods". In: *Journal of the American Statistical association* 66.336, pp. 846–850.

Raykov, Y. P., A. Boukouvalas, and M. A. Little (2014). "Simple approximate MAP Inference for Dirichlet processes". In: *arXiv preprint arXiv:1411.0939*.

Richardson, S. and P. J. Green (1997). "On Bayesian analysis of mixtures with an unknown number of components (with discussion)". In: *Journal of the Royal Statistical Society: series B (statistical methodology)* 59.4, pp. 731–792.

Robert, C. P., T. Ryden, and D. M. Titterington (2000). "Bayesian inference in hidden Markov models through the reversible jump Markov chain Monte Carlo method". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 62.1, pp. 57–75.

Roeder, K. (1990). "Density estimation with confidence sets exemplified by superclusters and voids in the galaxies". In: *Journal of the American Statistical Association* 85.411, pp. 617–624.

Stephens, M. (2000). "Dealing with label switching in mixture models". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 62.4, pp. 795–809.

Strehl, A. and J. Ghosh (2003). "Cluster ensembles—a knowledge reuse framework for combining multiple partitions". In: *The Journal of Machine Learning Research* 3, pp. 583–617.

Vinh, N. X., J. Epps, and J. Bailey (2009). "Information theoretic measures for clusterings comparison: is a correction for chance necessary?" In: *Proceedings of the 26th Annual International Conference on Machine Learning.* ACM, pp. 1073–1080.

Vinh, N. X., J. Epps, and J. Bailey (2010). "Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance". In: *Journal of Machine Learning Research* 11, pp. 2837–2854.

Wade, S. and Z. Ghahramani (2015). "Bayesian cluster analysis: Point estimation and credible balls". In: *arXiv preprint arXiv:1505.03339.*

White, A., J. Wyse, and T. B. Murphy (2016). "Bayesian variable selection for latent class analysis using a collapsed Gibbs sampler". In: *Statistics and Computing* 26.1-2, pp. 511–527.

Wyse, J. and N. Friel (2012). "Block clustering with collapsed latent block models". In: *Statistics and Computing* 22.2, pp. 415–428.

Wyse, J., N. Friel, and P. Latouche (2014). "Inferring structure in bipartite networks using the latent block model and exact ICL". In: *arXiv preprint arXiv:1404.2911.*

Zanghi, H., C. Ambroise, and V. Miele (2008). "Fast online graph clustering via Erdős–Rényi mixture". In: *Pattern Recognition* 41.12, pp. 3592–3599.

A   Appendix

A.1   A simplified allocation sampler

The methodology described throughout the paper requires a sample of partitions $\mathbf{Z}$. In practice, such sample may be obtained using Markov Chain Monte Carlo algorithms such as the Reversible Jump algorithm (Green 1995) and the Allocation Sampler (Nobile and Fearnside 2007). In fact, the distinctive feature of these samplers is that they can move across models, hence allowing Bayesian inference on the unknown number of groups. The adaptation of these algorithms to an arbitrary clustering context can be quite challenging. For this reason we describe in this appendix a very simple and general purpose allocation sampler, that can be applied in a wide range of clustering contexts.

The sampler can be thought of as a simplified version of the Allocation Sampler of Nobile and Fearnside (2007), where only one type of update step is used. The marginal posterior for the allocations, denoted by $\pi\left(\mathbf{z}|\mathcal{Y}\right)$, is assumed to be available in exact form, up to a proportionality constant. As required by the Metropolis-Hastings algorithm, a proposal distribution $q\left(\mathbf{z}'|\mathbf{z}\right)$ is introduced, denoting the probability of proposing the new partition $\mathbf{z}'$ when the current partition is $\mathbf{z}$. The proposal q is described in the following section. At each step, the current partition is changed into the new proposed one with probability

$$\alpha\left(\mathbf{z},\mathbf{z}'\right) = \min\left\{1,\ \frac{q\left(\mathbf{z}|\mathbf{z}'\right)\pi\left(\mathbf{z}'|\mathcal{Y}\right)}{q\left(\mathbf{z}'|\mathbf{z}\right)\pi\left(\mathbf{z}|\mathcal{Y}\right)}\right\} \tag{32}$$

or it is left unchanged otherwise. The discrete process so-obtained is a Markov chain over the space $\mathcal{Z}$. If the proposal distribution makes such process ergodic, then its stationary distribution corresponds to the marginal posterior distribution $\pi\left(\mathbf{z}|\mathcal{Y}\right)$.

A.2   Proposal distribution

In Nobile and Fearnside (2007) the authors introduce a novel proposal distribution specifically designed to sample allocations, and composed of several types of updates. In particular they use the so-called ejection/absorption steps that are meant to enhance the mixing of the chain. In our sampler we simplify such a proposal distribution essentially confining it to these two steps only.

Given the current partition $\mathbf{z}$, let $\underline{N} = \left\{N_1,\ldots,N_{K_{up}}\right\}$ be the vector of counts for the clusters, and define the sets $\mathcal{U} = \{g : N_g > 0\}$ and $\mathcal{E} = \{g : N_g = 0\}$. K_{up} is the maximum number of groups allowed and may be set equal to N. The proposal $\mathbf{z}' \sim q\left(\cdot|\mathbf{z}\right)$ is constructed in the following way:

- Select an outbound group g uniformly at random in $\mathcal{U}$.

- If $|\mathcal{U}| = 1$ or $|\mathcal{U}| = K_{up}$ select an inbound group h uniformly at random in $\mathcal{U} \setminus \{g\}$

- Else with probability 0.5 select h uniformly at random in $\mathcal{U} \backslash \{g\}$ or from $\mathcal{E}$ otherwise.

- Once g and h are chosen, set the number of observations r to be moved from g to h: r is drawn uniformly at random from $\{1, 2, \ldots, N_g\}$ if $N_h > 0$ or from $\{1, 2, \ldots, \lceil N_g/2 \rceil\}$ otherwise.

- The items $\mathcal{I} = \{i_1, \ldots, i_r\}$ that are being moved are chosen uniformly at random within group g.

The partition $\mathbf{z}'$ is then equal to $\mathbf{z}$ with the allocations of items in $\mathcal{I}$ changed to h.

The probability of proposing $\mathbf{z}'$ given $\mathbf{z}$ is given by:

$$q\left(\mathbf{z}'|\mathbf{z}\right) = Pr\left(g\right) Pr\left(h|g\right) Pr\left(r|g, h\right) Pr\left(\mathcal{I}|g, r\right). \tag{33}$$

Here follows the explicit formulation of each of the terms on the rhs of (33).

$$Pr\left(g\right) = \frac{1}{|\mathcal{U}|} \text{ for all } g \in \mathcal{U}. \tag{34}$$

If $|\mathcal{U}| = 1$ or $|\mathcal{U}| = K_{up}$ then

$$Pr\left(h|g\right) = \frac{1}{K_{up} - 1} \text{ for all } h \in \mathcal{U} \setminus \{g\}, \tag{35}$$

otherwise

$$Pr\left(h|g\right) = \begin{cases} \frac{1}{2(|\mathcal{U}|-1)} & \text{for all } h \in \mathcal{U}; \\ \frac{1}{2|\mathcal{E}|} & \text{for all } h \in \mathcal{E}; \end{cases} \tag{36}$$

As concerns r:

$$Pr\left(r|g, h\right) = \begin{cases} \frac{1}{N_g} & \text{for all } r \in \{1, \ldots, N_g\} \text{ if } N_h > 0; \\ \frac{1}{\lceil N_g/2 \rceil} & \text{for all } r \in \{1, \ldots, \lceil N_g/2 \rceil\} \text{ if } N_h = 0; \end{cases} \tag{37}$$

and

$$Pr\left(\mathcal{I}|g, r\right) = \frac{1}{\binom{N_g}{r}}. \tag{38}$$