Recent advances in latent variable models:
from finite mixtures to latent block models

Nial Friel

joint work with Pierre Latouche, Riccardo Rastelli and Jason Wyse

November, 2019. Bayes on the Beach.
Motivation

- Latent variable models abound in statistics: finite mixtures; stochastic block models and latent block models, to name a few.

- The statistical objectives are usually two-fold:

  1. Infer the number of components or groups or blocks.
  2. Estimate an optimal *partition*, eg, to allocate: observations to components (*mixtures*); actors to blocks (*SBM*); members to block type 1 and clubs to block type 2 (*LBM*).

- Both tasks are typically computationally challenging.

The motivation of this talk is to provide a computationally efficient framework for these tasks.
A generic context for latent variable models

- The **observed data** $X$ characterise the **attributes** or **profiles** of $N$ individuals (or items or nodes).

- Each individual has an associated **unobserved** cluster membership variable (the **allocation**) $z_i \in \{1, \ldots, K\}$, for $i = 1, 2, \ldots, N$.

- $z_i$ determines the **distribution** of the observations related to $i$.

- **Examples**: **Finite Mixtures**, Infinite Mixtures, Hidden Markov Models, **Block Models for networks**.
A Bayesian framework for latent variable models

\[ \pi(Z, \theta, \alpha | X, \phi, \delta, K) \propto f(X|Z, \theta) \pi(\theta|\phi) \pi(Z|\alpha) \pi(\alpha|\delta) \]
A general Bayesian clustering framework

We are interested in cases when parameters $\theta$ and $\alpha$ can be integrated out (collapsed), yielding a marginal posterior:

$$
\pi(Z|X, \phi, \delta) \propto \int_{\Theta} f(X|Z, \theta) \pi(\theta|\phi) \, d\theta \int_{A} \pi(Z|\alpha) \pi(\alpha|\delta) \, d\alpha \\
\propto f(X|Z, \phi) \pi(Z|\delta) \\
= f(X, Z|\phi, \delta).
$$

A sufficient condition to obtain this is that $\pi(\theta|\phi)$ and $\pi(\alpha|\delta)$ are conjugate priors to $f(X|Z, \theta)$ and $\pi(Z|\alpha)$, respectively.

In such case, $\pi(Z|X, \phi, \delta)$ has an exact explicit form.
The Integrated Completed Likelihood (ICL) is a model-based criterion that has had enormous impact. It aims to maximise \( f(X, Z|K) \).

It differs from BIC in that it makes use of the complete data to choose the number of groups:

- BIC approximates \( f(X|K) \).
- ICL approximates \( f(X, Z|K) = f(X|K) \pi(Z|X, K) \).

It turns out that ICL is equal to BIC with an additional penalisation term equal to the estimated mean entropy of the allocation vector \( Z \).

Partitions exhibiting well separated groups are preferred when ICL is used.
**Exact ICL criteria**

However, as a function of the allocation \( Z \), the marginal posterior satisfies:

\[
\pi(Z|X, \phi, \delta, K) \propto f(X, Z|\phi, \delta, K)
\]

which is the exact value that the ICL criterion approximates.

As a consequence, in the space of all possible partitions (any \( K \)) the solution of the optimisation problem will also maximise the exact ICL.
Applicability of this strategy

Note that this collapsing strategy can be implemented in a variety of different modelling frameworks:

1. **Finite mixtures**: Nobile and Fearnside, 2007), (Bertoletti, NF and Rastelli, 2015).
2. **Stochastic block models**: (McDaid et al, 2013), (Côme and Latouche, 2015), (Rastelli, Latouche, NF, 2018).

In essence, this implies that we can implement an exact ICL criteria in these (and other) contexts.
1. Dirichlet-Multinomial conjugacy is employed to provide an analytic expression for \( f(Z|\delta) \).

2. As concerns \( f(X|Z, \phi) \):

Define \( \theta = \{ \mathbf{m}_g, R_g \} \) and hyperparameters \( \phi = \{ \mu, \tau, \nu, \xi \} \) and consider the following hierarchical structure:

\[
\begin{align*}
\mathbf{x}_i | z_i = g, \mathbf{m}_g, R_g & \sim \text{MVN}_b \left( \mathbf{m}_g, R_g^{-1} \right) \\
\mathbf{m}_g | \mu, \tau, R_g & \sim \text{MVN}_b \left( \mu, [\tau R_g]^{-1} \right) \\
R_g | \nu, \xi & \sim \text{Wishart} \left( \nu, \xi \right)
\end{align*}
\]

It is straightforward to show that this yields an analytic expression for \( f(X|Z, \phi) \).

Both then combine to give an exact form for the complete model evidence \( f(Z, X|\phi, \delta) \).
Inference based on the marginal posterior $\pi(Z|X, \phi, \delta)$

Note that $K$ can be automatically inferred since $Z$ is categorical.

Inference on $Z$ can be performed through:

- **MCMC techniques** based on a collapsed Gibbs sampler:
  - The mixing properties of the sampler improve (Liu, 1994).
  - Model parameters can be recovered in a second stage.
  - See Nobile and Fearnside (2007) for details.

- **Discrete optimisation** of the marginal posterior with respect to $Z$, for every $K$:
  - Very fast.
  - No mixing and label switching issues to deal with.
  - Optimal $Z$ is MAP and the corresponding $K$ is optimal with respect to the exact ICL.
Optimisation of the exact ICL (Bertoletti et al, 2015)

Greedy algorithm:

1. Start with a random initial partition $Z^{(0)}$.

2. Reallocate each observation in turn (random order), moving it to the group that gives the best increase in exact ICL.

3. Repeat step 2 until no further increase is possible or max number of iterations is hit.

Features of the algorithm:

- Complexity of a single update does not depend on $N$.
- Convergence in very few iterations.
- High sensitivity to local optima.
Sensitivity to local optima: an example

\[ \text{ICL}_\text{ex} = -62.661 \]

\[ \text{ICL}_\text{ex} = -70.727 \]

\[ \text{ICL}_\text{ex} = -70.223 \]

\[ \text{ICL}_\text{ex} = -69.048 \]

\[ \text{ICL}_\text{ex} = -69.223 \]

\[ \text{ICL}_\text{ex} = -66.762 \]

\[ \text{ICL}_\text{ex} = -62.555 \]
Optimisation of the exact ICL

To avoid local optima, a **combined step** is proposed. Observations are reallocated in **blocks**, rather than singularly.

When observation \( i \) is selected:

1. **Draw** \( r \sim \text{BetaBinomial}(N_{z_i} - 1; b_0, b_1) \).

2. **Select** the \( r \) nearest neighbours of \( i \).

3. **Reallocate** all of these observations as a block to the best group.

*Euclidean distance* is used to select the nearest neighbours. Need an offline cost of \( \mathcal{O}(N^2) \).

Alternatively, neighbours can be chosen completely at random within the same group.
Simulated data Baudry et al. (2010)
Latent block models: Bipartite networks

Consider an observed bipartite network:

- Clubs: $1, \ldots, c$
- Members: $1, \ldots, m$

With adjacency matrix $X$ defined as

$$X_{ij} = \begin{cases} 
1, & \text{if member } i \text{ is in club } j; \\
0, & \text{otherwise.}
\end{cases}$$

Note: This model can also accommodate non-binary edges.
Bi-partite networks

Movie−Lens data

Movie−Lens data: 943 users 1682 movies

Movies rated/ not rated

Movies → clubs
Users → members
Bi-partite networks

Here the key questions/tasks are:

- Is there a co-clustering of members and clubs? If so, how many clusters for members and how many for clubs?

- Identify groups of members with similar linking attribute to groups of clubs, should they exist and vice-versa.
Latent block models (Govaert & Nadif, 2008)

- Assume there are $K$ member groups (rows), $G$ club groups (columns).

- For a member $i$ in group $k$, the linking attribute to club $j$ in group $g$ is modelled by
  $$f(X_{ij} | \theta_{kg}).$$

- Here we assume binary links (although this can extend to count links, continuous links etc).
  $$f(X_{ij} | \theta_{kg}) = \theta_{kg}^{X_{ij}} (1 - \theta_{kg})^{1 - X_{ij}}$$
Consider a generative model for $X_{ij}$:

- Label $z_i$ generated from $(1, \ldots, K)$ with weights $(\omega_1, \ldots, \omega_K)$

- Label $w_j$ generated from $(1, \ldots, G)$ with weights $(\rho_1, \ldots, \rho_G)$

- Conditioning on $z_i$ and $w_j$, $X_{ij}$ is generated from the model for links with parameter $\theta_{z_i w_j}$
  $$X_{ij} \sim f(\cdot|\theta_{z_i w_j}).$$
Latent block model

Let $z$ be a label vector such that $z_i = k$ if row (user) $i$ is in row group $k$. Similarly let $w_j$ be labels for the columns (movies) $j$.

The likelihood of observing the adjacency matrix $X$ can be written as a sum over all latent partitions

$$f(X|K, G, \theta, \omega, \rho) = \sum_{(z, w) \in \mathcal{Z} \times \mathcal{W}} \pi(z, w|\omega, \rho, K, G)f(X|z, w, \theta, K, G).$$

However, this is intractable, so as for the finite mixture case, we work with the likelihood completed with labels.
Latent block model

Assume row and column allocations are independent a priori:

\[
\pi(z, w|\omega, \rho, K, G) = \pi(z|\omega, K) \pi(w|\rho, G)
\]

\[
= \left( \prod_{i=1}^{m} \prod_{k=1}^{K} \omega_k^{I(z_i=k)} \right) \left( \prod_{j=1}^{c} \prod_{g=1}^{G} \rho_g^{I(w_j=g)} \right)
\]

Assume local independence of the \(X_{ij}\)'s conditioning on the labels

\[
f(X|z, w, \theta, K, G) = \prod_{k=1}^{K} \prod_{g=1}^{G} \prod_{i:z_i=k} \prod_{j:w_j=g} f(X_{ij}|\theta_{kg})
\]
Latent block model: conjugate priors

- \( \pi(\omega|K) \rightarrow \text{Dir}(\alpha, \ldots, \alpha). \) \((\text{Multinomial-Dirichlet})\)

- \( \pi(\rho|G) \rightarrow \text{Dir}(\beta, \ldots, \beta). \) \((\text{Multinomial-Dirichlet})\)

- Assume that \( \pi(\theta|K, G) \) is fully conjugate to \( f(X_{ij}|\theta_{kg}) \) i.e. integrating out \( \omega, \rho \) and \( \theta \) analytically. \((\text{Bernoulli-Beta})\)

Note that this choice of conjugate priors was also used in Wyse & Friel (2012) who used MCMC to infer \( z, w \) and \( K, G \).

It is also worth pointing out that McDaid et al. (2013) followed a similar strategy, but for the stochastic block model.
Exact Integrated completed likelihood

Consider the integrated complete data log likelihood giving rise to the ICL criterion

\[
\log \pi(X, z, w | K, G) = \log \left( \int_{\omega, \rho, \theta} \pi(X, z, w, \omega, \rho, \theta | K, G) \, d\omega \, d\rho \, d\theta \right)
\]

\[
= \log f(X|z, w, K, G) + \log \pi(z, w | K, G)
\]

\[
= \text{ICL}_{ex}(z, w, K, G)
\]

where

\[
\log f(X|z, w, K, G) = \log \left( \int_{\theta} f(X|z, w, \theta, K, G) \pi(\theta | K, G) \, d\theta \right)
\]

\[
\log \pi(z, w | K, G) = \log \left( \int_{\rho, \omega} \pi(z, w | \omega, \rho, K, G) \pi(\omega, \rho | K, G) \, d\rho \, d\omega \right)
\]
The exact ICL criterion can be used for selecting the number of clusters $K$ and $G$.

Côme and Latouche (2015) use a greedy search on the exact ICL to find the number of stochastic blocks as well as stochastic block memberships.

The advantage of the exact ICL approach is that it provides a tractable alternative to competing MCMC schemes. MCMC can have poor mixing and require very large number of iterations for big networks, resulting in infeasibly large run times.
ICL greedy search

- The scheme we use is applied alternately to rows and columns of our adjacency matrix.

- Firstly initialize the labels $z, w$ choosing conservative (larger than needed) values for $K$ and $G, K_{\text{max}}, G_{\text{max}}$.

- The greedy search algorithm iteratively allocates members and clubs and merges existing clusters so as to maximize the ICL.
Exact ICL greedy search

- Randomly scan the rows. Take member $i$ with $z_i = k$. Compute the change in ICL for moving member $i$ to cluster $l \neq k$

\[
\Delta_{k \rightarrow l} = \text{ICL}(z^*, w, K, G) - \text{ICL}(z, w, K, G)
\]

and we take $\Delta_{k \rightarrow k} = 0$.

- Move member $i$ to the cluster $l$ that gives the largest change in ICL. If all $\Delta_{k \rightarrow l}$ are negative, leave $i$ where it is.
Exact ICL greedy search

- If taking member $i$ from cluster $k$ would leave it empty, we compute the differences as

$$\Delta_{k \rightarrow l} = \text{ICL}(z^*, w, K - 1, G) - \text{ICL}(z, w, K, G).$$

- This is the process by which clusters disappear as the greedy search progresses.

- The process just described is applied to the clubs too.

- The greedy search terminates when no further moves can increase the ICL.
Strategy 1: Greedy search pruning

- After a few full sweeps of the data, we may already expect a good deal of clustering.

- Updating each row requires $O(c_M KG)$ calculation with $c_M$ average cost of computing a marginal likelihood.

- Reduce this cost by pruning/removing unlikely clusters.

- Low probabilities of being reassigned from cluster $k$ to $l$ correspond to large negative differences in exact ICL.
Strategy 1: Greedy search pruning

- For rows, the form of the full conditional for row label $i$ can be written

$$
\pi(z_i = k'|\text{everything else}) = \frac{\exp\{\Delta_k \rightarrow k'\}}{\sum_{l=1}^{K} \exp\{\Delta_k \rightarrow l\}}.
$$

where $k$ is the allocation of row $i$ from the previous iteration.

- Of most interest is when $\pi(z_i = k'|\text{everything else})$ is large compared with other groups i.e.

$$
\pi(z_i = k'|\text{everything else}) > 1 - \delta
$$

with $\delta$ small $\Rightarrow$ strong cohesion to group $k'$
Strategy 1: Greedy search pruning

Prune/remove clusters with a very small full conditional probability compared with cluster $k'$ where $k'$ gives the maximum change in ICL (can be the same as $k$).

Consider clusters pairwise

\[
\frac{\exp\{\Delta_{k \rightarrow k'}\}}{\exp\{\Delta_{k \rightarrow k'}\} + \exp\{\Delta_{k \rightarrow l}\}} > 1 - \delta
\]

or equivalently

\[
\Delta_{k \rightarrow k'} - \Delta_{k \rightarrow l} > \log \left[ \frac{1 - \delta}{\delta} \right]
\]

then prune cluster $l$ from the search options in future iterations.

Take $\log \left[ (1 - \delta)/\delta \right] = 150$. This implies very small $\delta$. 
Strategy 2: Sparse storage

- Store only the present ties and their locations.

- This turns out to be a useful strategy for sparse networks.

- We can then make a calculation to reduce vastly computations on the no-zero valued ties.
Applications – four algorithms

There are four possible algorithms available to us:

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Pruning</th>
<th>Sparse form</th>
</tr>
</thead>
<tbody>
<tr>
<td>A0</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>A1</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>A2</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>A3</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

In terms of speed A3 should be the fastest and A0 should be the slowest for large data.
Applicability beyond binary edges

Depending on the type of ties in the observed network, one has a choice of assumed models that still allow the ICL to be computed exactly.

| $f(X_{ij}|\theta_{kg})$ | $\pi(\theta_{kg})$ |
|------------------------|-------------------|
| Binomial               | Beta              |
| Multinomial            | Dirichlet         |
| Poisson                | Gamma             |
| Gaussian               | Gaussian-Gamma    |

This allows for probabilistic modelling of richer network information than tie/no-tie.
Applications – congressional voting

Application of the greedy exact ICL search to the UCI congressional voting data analysed in Wyse and Friel (2012) (abstain=nay for our purposes)

435 congressmen (members) voting on 16 key issues (clubs).

Number of groups found $K = 6$, $G = 11$.

Little difference between four algorithms (speed & max ICL).
Applications- congressional voting

Closer look at the randomness introduced by randomly processing rows.

100 runs of the algorithm gave the maximum ICL's reached

Algorithm run times averaged at 0.6 of a second.

This is in stark contrast to the 1 hour it took Wyse and Friel (2012) algorithm to generate 100,000 posterior samples of the allocation vectors.
Congressional voting data

Confusion matrix comparing the political affiliation with the Optimal partition.

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Democrat</td>
<td>42</td>
<td>79</td>
<td>127</td>
<td>16</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Republican</td>
<td>4</td>
<td>6</td>
<td>0</td>
<td>125</td>
<td>30</td>
<td>3</td>
</tr>
</tbody>
</table>
Movie-Lens data:
943 users
1682 movies

Movies rated/ not rated

Movies → clubs
Users → members
Applications - Movie-Lens 100k data

Each of the four algorithms A0-A3 were started with same random seed, allowing for a direct comparison.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>maximum ICL</th>
<th>time (sec)</th>
<th>(K, G)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A0</td>
<td>-225670.9</td>
<td>183.8</td>
<td>(49,40)</td>
</tr>
<tr>
<td>A1</td>
<td>-225670.9</td>
<td>69.8</td>
<td>(49,40)</td>
</tr>
<tr>
<td>A2</td>
<td>-225670.9</td>
<td>134.3</td>
<td>(49,40)</td>
</tr>
<tr>
<td>A3</td>
<td>-225670.9</td>
<td>51.8</td>
<td>(49,40)</td>
</tr>
</tbody>
</table>

All algorithms converge to the same solution from the same starting position. However, we see marked speed-up for using sparse forms (A1 & A3) and pruning (A2 & A3). Pruning can give faster run with a looser threshold, but this can introduce error.
Applications- Movie-Lens 100k data

Re-ordered matrix

Identified 49 user and 40 movie clusters.

MCMC is practically infeasible for even this size of matrix.
Optimal a posteriori partition via loss functions
Motivation

Consider a latent variable clustering framework, eg, finite mixtures, where a posterior sample of partitions (or allocations) is available:

How can one summarise the posterior sample?

What is an optimal partition?
Bayesian decision theory

Define a **loss function**:

\[ \mathcal{L} : \mathcal{Z} \times \mathcal{Z} \rightarrow \mathbb{R} \]

such that \( \mathcal{L}(A, Z) \) quantifies the loss occurring when the decision (estimator) is \( A \) while the true parameter is \( Z \).

\( A^* \in \mathcal{Z} \) is a **Bayes decision rule** if it minimises the expected posterior loss, defined as:

\[ \psi(A) := \mathbb{E}_{Z|X, K} [\mathcal{L}(A, Z)] = \sum_{Z \in \mathcal{Z}} \mathcal{L}(A, Z) \pi(Z|X, \phi, \delta). \]
Loss functions in clustering

Possible choices of $\mathcal{L}$ include:

- **0-1 loss**: $\mathcal{L}(A, Z) = 1_{\{A \neq Z\}} \Rightarrow A^*$ is MAP estimator.

- **Quadratic loss**: $A^*$ is the posterior mean.

However, in the **clustering context** described, such losses are arguably not sensible choices.

We propose instead two losses specifically designed to compare partitions:

- **Binder’s loss** (rescaled Rand index).

- **Variation of information loss** (derived from information theory).
Loss functions based on the contingency table

Consider the contingency table (or confusion matrix) for two partitions \( a \) and \( z \) with \( K_a \) and \( K_z \) groups, respectively.

<table>
<thead>
<tr>
<th>Classes</th>
<th>( 1 )</th>
<th>( \ldots )</th>
<th>( k_z )</th>
<th>( \sum )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( n_{11}^{a,z} )</td>
<td>( \ldots )</td>
<td>( n_{1,k_z}^{a,z} )</td>
<td>( n_1^a )</td>
</tr>
<tr>
<td>( \vdots )</td>
<td>( \vdots )</td>
<td>( \vdots )</td>
<td>( \vdots )</td>
<td>( \vdots )</td>
</tr>
<tr>
<td>( k_a )</td>
<td>( n_{k_a,1}^{a,z} )</td>
<td>( \ldots )</td>
<td>( n_{k_a,k_z}^{a,z} )</td>
<td>( n_{k_a}^a )</td>
</tr>
<tr>
<td>( \sum )</td>
<td>( n_{1}^{z} )</td>
<td>( \ldots )</td>
<td>( n_{k_z}^{z} )</td>
<td>( N )</td>
</tr>
</tbody>
</table>

with entry \( gh \) equal to

\[
n_{gh}^{a,z} = \sum_{i=1}^{N} \mathbb{1}_{\{a_i=g\}} \mathbb{1}_{\{z_i=h\}}
\]
Entropy based loss functions

▶ The entropies of $a$ and $z$:

$$H(a) = - \sum_{g=1}^{K_a} \frac{n^a_g}{N} \log_2 \frac{n^a_g}{N}; \quad H(z) = - \sum_{h=1}^{K_z} \frac{n^z_h}{N} \log_2 \frac{n^z_h}{N}. \quad (1)$$

▶ The joint entropy of $a$ and $z$:

$$H(a, z) = - \sum_{g=1}^{K_a} \sum_{h=1}^{K_z} \frac{n^{a,z}_{gh}}{N} \log_2 \frac{n^{a,z}_{gh}}{N}. \quad (2)$$

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

$$I(a, z) = H(a) + H(z) - H(a, z). \quad (3)$$
The Variation of Information (VI) loss is defined as:

$$\mathcal{L}_{VI} (a, z) = 2H(a, z) - H(a) - H(z).$$

or equivalently as

$$\mathcal{L}_{VI} (a, z) = H(a, z) - I(a, z).$$
Properties of the VI loss

- **Metric**: the VI loss satisfies the axioms for a metric.

- **Invariant to label permutations** so respects exchangeability.

- **Local property**: if the partition $a$ is obtained from $z$ by splitting one cluster into two, then $\mathcal{L}_{VI}(a, z)$ depends only on the cluster undergoing the split.

- **Upper and lower bounds**: the VI loss is always smaller or equal to $\log N$, and $\mathcal{L}_{VI}(a, z) = 0$ iff $a \equiv z$. Also, if $\mathcal{L}(a, z) \neq 0$ then $\mathcal{L}_{VI}(a, z) \geq \frac{2}{N}$.

All of these properties and more can be found in Meilă (2007).
Binder’s loss function

Binder’s loss can be written as:

\[ \mathcal{L}_B (a, z) = \sum_{i<j} \left\{ \mathbb{1}\{a_i=a_j\} \mathbb{1}\{z_i \neq z_j\} + \mathbb{1}\{a_i \neq a_j\} \mathbb{1}\{z_i=z_j\} \right\} . \]

This penalises the loss resulting from allocating two observations to the same (different) cluster when they should be in different (the same) cluster.

Or equivalently as,

\[ \mathcal{L}_B (a, z) = \frac{1}{2} \sum_{g=1}^{K_a} (n^a_g)^2 + \frac{1}{2} \sum_{h=1}^{K_z} (n^z_h)^2 - \sum_{g=1}^{K_a} \sum_{h=1}^{K_z} (n^{a,z}_{gh})^2 . \]
Minimisation of expected loss

A posterior sample $Z^{(1)}, \ldots, Z^{(T)} \sim \pi(\cdot | X, \phi, \delta)$ can be obtained in several ways:

- **Allocation sampler** of Nobile and Fearnside 2007.
- Ad-hoc **collapsed Gibbs samplers**.

Approximation of expected posterior loss:

$$\psi(A) \approx \frac{1}{T} \sum_{t=1}^{T} \mathcal{L}(A, Z^{(t)}).$$

We are developing a tool that can effectively minimise the expected posterior loss.
Simulated data

True partition

MAP

Binder's loss

VI loss
Old Faithful Geyser Data

BIC

MAP

Binder's loss

VI loss
Application to the French political blogosphere

- The data consist of a **network** of 196 nodes and 2864 edges.

- Nodes represent blogs’ **hostnames**, and edges **hyperlinks** between webpages.

- **Political affiliations** are **known**: 11 different political parties are present.

- The **main political parties** are:
  - UMP (French “republican”);
  - UDF (“moderate” party);
  - Liberals (supporters of economic liberalism);
  - PS (French “democrat”).
An algorithm presented in (McDaid et al 2013) was used to draw posterior allocations.

A sample of 10,000 was collected, thinning by 100.

100 runs of the greedy algorithm was performed. The optimal value was reached in 10 runs.

CPU times: MCMC sampling: 5 hours; Greedy algorithm: 50 seconds per run.

We also implemented a variational algorithm implemented in mixer to obtain a second partition for reference.
French political blogs’ network

**Posterior distribution for the number of groups**

![Posterior distribution graph]

**French blogs: EPL values**

![EPL values graph]
French political blogs’ network

French political blogs
variational action

11 groups

French political blogs
VI Bayes action

18 groups
Conclusions/Further work

▶ The exact ICL approach is applicable to other latent variable contexts, eg, SBMs, hidden Markov models etc. It is worthwhile to investigate these contexts.

▶ Although the exact ICL approach is vastly more efficient than, eg, MCMC. It would highly desirable to develop MCMC strategies to efficiently sample allocations. This in turn would be important for estimating Bayes loss estimators.

▶ Further work is needed to improve scalability by exploiting sparsity and other ideas.

▶ Convergence results for greedy exact ICL search and investigation of other search strategies would be desirable.

▶ Finally, one could validly ask is exact ICL better (in terms of clustering and model choice) than the usual approximate ICL. This needs to be investigated.
Bertoletti, Friel, and Rastelli (2015) Choosing the number of clusters in a finite mixture model using an exact integrated completed likelihood criterion. *Metron*.


