Have gender stereotypes in words changed through the 20th century?

Brassoli Lorenzo
Ciceri Federico
Pigani Giovanni
Rota Francesca

17th February 2021

Contents

1. Introduction

This report is aimed at showing our results of the project of the course Bayesian Statistics.
The goal of this project is to develop and make more efficient a MCMC algorithm computing the posterior distribution in a Bayesian nonparametric dynamic mixture model (De Iorio, Guglielmi, Favaro, Ye, 2019) for clustering subjects over time.

The project is about dynamic clustering, where clustering configuration at time $t + 1$ depends on the previous one at time t. A Bayesian non-parametric approach to dynamic clustering via mixture modeling is taken. Moreover this model is applied to the study of the temporal dynamics of gender stereotypes in adjectives and occupations in the 20th and 21st centuries in the United States.

In the first part of the project we wanted to better understand how the model is implemented. We focused on how an Autoregressive Dirichlet Process and a particle MCMC are structured.
In the second part of the project we studied different clustering methods such as k-means or PAM and we compared the results with different indices for further evaluations.
In the third part of the project we focused on the optimization of the code in C++ and we presented our results.

2. The Model

In the first part of the project we studied the main theoretical characteristics of our model.

2.1. Dataset

In order to answer if gender stereotypes in words changed during time, we studied the change over time in in the 20th and 21th centuries in the United States of words as adjectives and occupations. [1] The aim is to merge different words into "gender" clusters (men, women and neutral words). In order to classify words associated to men and women, we used word embeddings provided by Garg et al. [2018] which measured the gender bias [2]. We obtained data for standardized adjectives and occupations' biases for women for each word. In particular a negative value for the bias means that the embedding more closely associates the (occupation or adjective) word with men, because the distance between the word is closer to men than women. Hence, gender bias corresponds to either negative or positive values of the embedding bias.

Dataset: Y_{jt}= embedding bias of word j at time t

$$\mathrm{Y}_{jt}= \begin{cases} \text{bias against women} \rightarrow \text{male word} & \text{if } Y_{jt} < 0 \\ \text{bias in favor of women} \rightarrow \text{female word} & \text{if } Y_{jt} > 0 \end{cases}$$

where t are the decades of the 20th century.

2.2. Bayesian non-parametric dynamic mixture model

In order to study how gender bias changes during time, we adopted a bayesian non-parametric dynamic mixture model proposed by De Iorio, Favaro, Guglielmi and Ye because it is a powerful tool for the analysis of time-dependent data. The model tries to describe the change of clustering structure over time: clustering configuration at time $t+1$ depends on the previous one at time t.
The degree of dependence is covered by a AR1-DP model.

2.2.1. Autoregressive Dirichlet process

Firstly we introduce the main objects used to build our model.
In order to describe the flexibility of the clustering dynamics of our occupations and adjectives during time we introduce a discrete time stochastic process $\epsilon= (\epsilon_t)_{t\geqslant 1}$ defined as follows:

$$\epsilon_1 \sim N(0,1) \qquad \epsilon_t = \psi\epsilon_{t-1} + \eta_t \quad t \geqslant 2$$

where $\psi \in (-1,1)$ and $(\eta_t)_{t>1}$ are i.i.d as $N(0, 1-\psi^2)$.
The process ϵ is also known as an Autoregressive stochastic process of order 1 with parameter ψ ($\epsilon \sim AR(1;\psi)$ in short).
The dependence between clustering structure from a decade to the following one can be well modeled by considering, a sequence of i.i.d autoregressive processes $(\epsilon_l)_{l\geqslant 1}$ where $\epsilon_l \sim (\mathrm{AR1};\psi)$ as follows:

$$\xi_{tl} = F^{-1}(\Phi(\epsilon_{tl}); a, b) \qquad t \geqslant 1, l \geqslant 1$$

where Φ denotes the CDF of standard gaussian distribution and $F(\cdot,;a,b)$ is the CDF of a beta random variable of parameters (a,b).

[1] See the Appendix for the Dataset.
[2] GitHub page at https://github.com/nikhgarg/EmbeddingDynamicStereotypes.

As a consequence ξ_{tl} depends on $\xi_{(t-1)l}$ with respect to the dependence parameter ψ, hence $\psi = 0$ means independence among ξ_{tl}. Furthermore, the $(\xi_{tl})_{t\geqslant 1, l\geqslant 1}$ so defined are beta random variables of parameters (a, b).

In order to describe the clustering dynamics an AR1-DP model is adopted.
Let $(G_t)_{t>1}$ a sequence of dependent random probability measures (RPMs), with t discrete, such that $G_t \sim DP(M, G_0)\ \forall t$, where $M > 0$ is the total mass parameter and G_0 denotes a non-atomic distribution on $\mathbf{\Theta}$

Stick Breaking
Thanks to Sethuraman's theorem, we can consider the stick breaking construction of a Dirichlet Process.
Let $(\xi_{tl})_{t\geqslant 1, l\geqslant 1} \overset{\text{iid}}{\sim} Beta(a, b)|_{a=1, b=M}$ independent from $(\theta_h)_{h\geqslant 1} \overset{\text{iid}}{\sim} G_0$.
Consider the following weights of the stick breaking built with the sequence ξ_{tl} as

$$w_{t1} = \xi_{t1}, \qquad w_{th} = \xi_{th} \prod_{l=1}^{h-1} (1 - \xi_{tl}) \quad h \geqslant 2$$

Then it follows that

$$G_t = \sum\nolimits_{h\geqslant 1} w_{th} \delta_{\theta_h} \tag{2.1}$$

The sequence $(G_t)_{t\geqslant}$ is called Autoregressive Dirichlet Process of order 1, in short $(G_t)_{t\geqslant 1} \sim$AR1-DP$(\psi, M, G_0)$

2.2.2. Autoregressive Dirichlet process mixture model

Since our data have continuous representation, a sequence of dependent RPMs is not enough to model them due to the fact that realizations of a discrete RPM are a.s discrete random variables.
For this reason an Autoregressive Dirichlet Process Mixture model is needed, because it represents the combination of a mixture of gaussian parametric distributions, where parameters are realizations of an AR1-DP.
We can describe the occupations and adjectives' biases y_{tj} with an AR1-DP mixture model with Gaussian kernel as follows:

$$\begin{aligned}
Y_{tj}|(\mu_{tj}, \tau_{tj}) &\overset{\text{ind}}{\sim} N(\mu_{tj}, (\lambda\tau_{tj})^{-1}) \qquad j = 1, \ldots, n \\
(\mu_{tj}, \tau_{tj})|G_t &\overset{\text{i.i.d}}{\sim} G_t \qquad t = 1900, 1910, \ldots, 2000 \\
(G_t)_{t\geqslant 1} &\sim AR1 - DP(\psi, M, G_0) \\
G_0 &\sim Gaussian - Gamma(\mu_0, \lambda, \alpha, \beta) \\
M &\sim Gamma(\alpha_M, \beta_M) \\
\psi &\sim TN(\mu_\psi, \sigma^2_\psi, -1, 1)
\end{aligned}$$

Where $\mu_0 = 0$, $\lambda = 0.01$, $\alpha = 2$, $\beta = 1$, $\alpha_{M_{occ}} = 4$, $\beta_{M_{occ}} = 4$, $\alpha_{M_{adj}} = 3$, $\beta_{M_{adj}} = 5$. The prior on M implies that the expected number of clusters at time t for occupations is 4 while for adjectives is 5.
Finally $TN(\cdot, \cdot, b\_lim, u\_lim)$ denotes a truncated gaussian distribution and we set $\mu_\psi = 0$ and $\sigma^2_\psi = 0.09$.

2.3. MCMC

We now need a strategy to sample from the posterior distribution of a AR1-DP mixture model. Since the stick breaking representation relies on an infinite series (2.1), we have to consider a truncation to J. Hence we augment the problem by introducing the allocation variables $\boldsymbol{s}_t$ for each latent variable $\boldsymbol{\theta}_t$, so that datum Y_{tj} is allocated to the component s_{tj} of the mixture at time t.
The model representation in latent form is

$$\begin{aligned} Y_{tj}|s_{tj}, \theta_{tj} &\overset{\text{ind}}{\sim} \mathcal{N}(\cdot; \theta_{s_{tj}}) \qquad j = 1, \dots, n \\ s_{tj}|\mathbf{w}_t &\overset{\text{i.i.d}}{\sim} \sum_{h \geqslant 1}^{J} w_{th}\delta_h \qquad j = 1, \dots, n \\ \theta_{tj} &\overset{\text{i.i.d}}{\sim} G_0. \end{aligned}$$

where $\theta_{tj} = (\mu_{tj}, \tau_{tj})$.
Now let l be the index corresponding to the unique non-zero term in the summation for s_{tj}. Consequently, for ease of notation, we will hereafter refer to θ_{tj} as θ_l, if $s_{tj} = l$.

The following MCMC scheme is designed to sample from the posterior distributions of $\{\boldsymbol{s}_t, \boldsymbol{\theta}, M, \psi, \boldsymbol{\epsilon}_t, \mathbf{w}_t\}$, i.e. the unknown parameters of the model:

- sampling $\boldsymbol{s}_t$ given the rest: since the allocations at different times are independent given the corresponding $\boldsymbol{w}_t$, the joint distribution of $(\boldsymbol{s}_1, \dots, \boldsymbol{s}_T)$ can be factorized and we can sample each allocation accordingly:

  $$[s_{tj} \mid \text{rest}] \;\propto\; w_{tl} f\left(y_{tj}, \theta_l\right)$$

  In particular, at time t the new allocation for occupation j is drawn from a discrete distribution with support $\{1, \dots, J\}$ and from weights proportional to w_{tj}. Each weight is associated to the latent variable θ_l, and to $f_{\mathcal{N}\left(\mu_l, \frac{1}{\tau_l}\right)}(y_{tj})$, i.e. the likelihood to observe the current datum assuming it is generated according to the Gaussian distribution with θ_l as representative.

- sampling $\boldsymbol{\theta}$ given the rest: we sample the non-allocated values of $\boldsymbol{\theta}$ from the base distribution: $\theta_l \overset{\text{i.i.d}}{\sim} G_0$. The allocated values, instead, are updated according to:

  $$[\theta_l \mid \text{rest}] \;\propto\; G_0\left(d\theta\right) \prod_{(t,j):s_{tj}=l} f\left(y_{tj}; \theta_{tj}\right)$$

Thus, assuming a conjugate Gaussian-Gamma prior for $\boldsymbol{\theta}$.

$$\tau_l \mid\sim Gamma(\alpha_0^{(\tau)}, \beta_0^{(\tau)})\mu_l \mid \tau_l \sim N(\mu_0, \frac{1}{\lambda_0\tau_l})$$

the update of the posterior hyperparameters is trivially computed:

$$\begin{gathered}\tau_l \mid \boldsymbol{Y}_l \sim Gamma(\alpha_{n_l}^{(\tau)}, \beta_{n_l}^{(\tau)}) \\ \mu_l \mid \tau_l \sim N(\mu_{n_l}, \tfrac{1}{\lambda_{n_l}\tau_l}) \\ \boldsymbol{Y}_l := \{Y_{tj},\ t \in \{1,\dots,T\},\ j \in \{1,\dots,N\} :\ s_{tj} = l\}\end{gathered}$$

where $n_l =\mid \boldsymbol{Y}_l \mid$, $\bar{y}_l = \frac{1}{n_l}\sum_{y_{tj}\in\boldsymbol{Y}_l} y_{tj}$, $s_l = \frac{1}{n_l}\sum_{y_{tj}\in\boldsymbol{Y}_l}(y_{tj}-\bar{y}_l)^2$;
$\mu_{n_l} = \frac{\lambda_0\mu_0+n_l\bar{y}_l}{\lambda_0+n_l}$, $\lambda_n = \lambda_0 + n_l$, $\alpha_{n_l}^{(\tau)} = \alpha_0^{(\tau)} + \frac{n_l}{2}$, $\beta_{n_l}^{(\tau)} = \beta_0^{(\tau)} + \frac{n_l sl}{2} + \frac{\lambda_0 n_l(\bar{y}_l-\mu_0)}{2(\lambda_0+n_l)}$.

- updating M given the rest: priorly assuming $M \sim Gamma\,(\alpha_M, \beta_M)$, we sample the updated value $M^{(i)}$ from a mixture of Gamma distributions:

$$\begin{aligned}M^{(i)} \mid \text{rest} \ \sim\ & \pi_\eta Gamma(\alpha_M + k^{(i)}, \beta_M - log\eta^{(i)}) + \ \dots \\ \dots\ & + (1-\pi_\eta)\, Gamma(\alpha_M + k^{(i)} - 1, \beta_M - log\eta^{(i)})\end{aligned}$$

with $\eta^{(i)} \sim Beta\left(M^{(i-1)}+1, TN\right)$; $\pi_\eta := \frac{\chi_\eta}{1+\chi_\eta}$; $\chi_\eta := \frac{\alpha_M+k^{(i)}-1}{TN\left(\beta_M - log\eta^{(i)}\right)}$;
$k^{(i)}$ denotes the number of different clusters in $\boldsymbol{s}_t^{(i)}$; i represents the current iteration of the algorithm.

- sampling (ψ,$\boldsymbol{\epsilon}_t$,$\mathbf{w}_t$) given the rest: here we take advantage of a step of a particle MCMC sampler. In the following subsection we are going to introduce the sampling scheme of $[\psi, \boldsymbol{\epsilon}_t, \mathbf{w}_t \mid \boldsymbol{s}_t]$: we will focus on the joint update of ψ and $\boldsymbol{\epsilon}_t$ given the rest, since we can exploit the fact that: $\mathcal{L}_\psi\left(\mathbf{w}_t \mid \mathbf{w}_{t-1}\right) = \mathcal{L}_\psi\left(\boldsymbol{\epsilon}_t \mid \boldsymbol{\epsilon}_{t-1}\right)$.
The update of $\mathbf{w}_t$ is straightforward, since it suffices to compute the stick breaking weights given $\boldsymbol{\epsilon}_t$.

2.3.1. Particle MCMC

Particle Markov Chain Monte Carlo (pMCMC) is a stochastic algorithm designed to generate samples from a probability distribution which does not admit a closed form expression.
The idea behind particle MCMC methods is that the output of a sequential Monte Carlo algorithm is used as a proposal distribution to feed standard MCMC techniques.

——————————————— In our specific case of study, the target of the algorithm is the following joint conditional distribution:

$$p\left(\psi, \boldsymbol{\epsilon}_{1:T} \mid \boldsymbol{s}_{1:T}\right).$$

The design of a simulation pattern that allows to sample alternatively from the full-conditionals seems adequate here:

– $\psi^{(i)} \sim p\left(\cdot \mid \boldsymbol{\epsilon}_{1:T}^{(i-1)}\right)$

– $\boldsymbol{\epsilon}_{1:T}^{(i)} \sim p_{\psi^{(i)}}\left(\cdot \mid \boldsymbol{s}_{1:T}^{(i)}\right)$[3]

once $\boldsymbol{\epsilon}_{1:T}^{(0)}$ and $\psi^{(0)}$ are properly initialized.

The main difference with respect to a standard Monte Carlo approach is that, whenever a sample from $p_\psi\left(\boldsymbol{\epsilon}_{1:T} \mid \boldsymbol{s}_{1:T}\right)$ is needed, its SMC approximation comes into play. Among the two plausible PMCMC approaches in this context, namely particle Marginal Metropolis-Hastings sampler and particle Gibbs sampler, here we employ the former, though requiring the design of a proposal distribution for ψ.

Let us firstly describe the update of ψ: here a step of a Metropolis-Hasting sampler is employed. In particular, we assume a prior truncated Gaussian distribution for ψ, i.e. $\psi \sim \mathcal{TN}\left(0, 0.3^2, -1, 1\right)$; the proposal distribution is still a truncated Gaussian. The likelihood term involved in the computation of the acceptance ratio can be instead derived from (NUMERO EQUAZIONE AR1 ϵ).
Thus according to the following scheme, we are able to sample a new value of ψ:

- sample $\psi^\star \sim \mathcal{TN}\left(\psi^{(i-1)}, \left(\sigma_\psi^{(i-1)}\right)^2, -1, 1\right)$[4];
- set $\psi^{(i)} = \psi^\star$ with probability $\min\{1, \alpha\}$, where

$$\alpha = \frac{\pi\left(\psi^\star\right)}{\pi\left(\psi^{(i-1)}\right)} \frac{\prod_t p_{\psi^\star}\left(\boldsymbol{\epsilon}_t \mid \boldsymbol{\epsilon}_{t-1}\right)}{\prod_t p_{\psi^{(i-1)}}\left(\boldsymbol{\epsilon}_t \mid \boldsymbol{\epsilon}_{t-1}\right)} \frac{\mathcal{TN}(\psi^{(i-1)}; \psi^\star, (\sigma_\psi^{(i-1)})^2, -1, 1)}{\mathcal{TN}(\psi^\star; \psi^{(i-1)}, (\sigma_\psi^{(i-1)})^2, -1, 1)};$$

  otherwise set $\psi^{(i)} = \psi^{(i-1)}$.

Given the updated value of ψ, we now focus on sampling $\boldsymbol{\epsilon}_{1:T}$ given the rest.
Let us assume a Hidden Markov Model, i.e. a State Space Model in which the hidden state process $\{\boldsymbol{\epsilon}_t;\, t \geqslant 1\}$ cannot be observed directly, but only through another process: $\{\boldsymbol{s}_t;\, t \geqslant 1\}$. In such a setting, we are able to outline the Sequential Monte Carlo algorithm, which plays a central role in the Particle MCMC: in fact it consists of a sequential approximation of the posterior distribution: $p_\psi\left(\boldsymbol{\epsilon}_{1:T} \mid \mathbf{s}_{1:T}\right)$.

The approximation is achieved using a set of R weighted random samples called particles, indexed by r in the following.

The output of the SMC algorithm is

$$\hat{p}_\psi\left(\boldsymbol{\epsilon}_1, \ldots, \boldsymbol{\epsilon}_T \mid \mathbf{s}_1, \ldots, \mathbf{s}_T\right) := \sum_{r=1}^{R} \tilde{\omega}_T^r \delta_{\boldsymbol{\epsilon}_{1:T}^r},$$

where $\tilde{\omega}_T^r$ is the importance weight associated to $\boldsymbol{\epsilon}_{1:T}^r$.

[3] for ease of notation, the conditioning to ψ will hereafter be indicated with a sub-index.
[4] in order to keep a balanced proportion of acceptance, σ_ψ is occasionally adjusted every 200 iterations.

For each $t \geqslant 2$ we perform the so called resampling step: for a fixed particle r, we sample from a discrete probability distribution on $\{1, \dots, R\}$ according to weights $\left(\tilde{\omega}_{t-1}^r\right)_{1 \leqslant r \leqslant R}$. This operation can be interpreted as the step by which we assign to the offspring particle r at time t its ancestor particle A_{t-1}^r at time $t-1$. The introduction of these auxiliary variables allows to keep track of the history of the particles.
We can also define B_t^r as the index that the ancestor of an outcome particle r (i.e. associated to a complete episode: $\boldsymbol{\epsilon}_{1:T}^r$) had at time t. Exploiting the fact that $B_T^r = r$ and the recursive relation $B_t^r = A_{t+1}^{B_{t+1}^r}$, we can say that the algorithm designs a trajectory $\boldsymbol{\epsilon}_{1:T}^r = \left(\boldsymbol{\epsilon}_1^{B_1^r}, \dots, \boldsymbol{\epsilon}_T^{B_T^r}\right)$ associated with the history $B_{1:T}^r = (B_1^r, \dots, {}_T^r = r)$ of each particle r.
This is one of the crucial advantages of a sequential approach: in fact we can simply draw an index r from $\{1, \dots, R\}$ using $(\tilde{\omega}_T^r)_{1 \leqslant r \leqslant R}$ as weights, and set $\boldsymbol{\epsilon}_{1:T}^{(i)} = \boldsymbol{\epsilon}_{1:T}^r$ ($B_{1:T}^{(i)}$ is also implicitly sampled).

As in standard procedures, the computation of the importance weights at time $t = 1$ is performed according to:

$$\omega_1(\boldsymbol{\epsilon}_1^r) = \frac{p_\psi(\boldsymbol{\epsilon}_1^r, \boldsymbol{s_1})}{q_\psi(\boldsymbol{\epsilon}_1^r \mid \boldsymbol{s}_1)} = \frac{\mu_\psi(\boldsymbol{\epsilon}_1^r)\, g_\psi(\boldsymbol{s}_1 \mid \boldsymbol{\epsilon}_1^r)}{q_\psi(\boldsymbol{\epsilon}_1^r \mid \boldsymbol{s}_1)}$$

and for every $t \geqslant 2$:

$$\begin{aligned}\omega_t(\boldsymbol{\epsilon}_{1:t}^r) = \frac{p_\psi(\boldsymbol{\epsilon}_{1:t}^r, \boldsymbol{s}_{1:\boldsymbol{t}})}{p_\psi(\boldsymbol{\epsilon}_{1:t-1}^{A_{t-1}^r}, \boldsymbol{s}_{1:t-1})\ q_\psi(\boldsymbol{\epsilon}_{1:t}^r \mid \boldsymbol{s}_{1:t}, \boldsymbol{\epsilon}_{1:t-1}^{A_{t-1}^r})} = \\ \frac{p_\psi(\boldsymbol{\epsilon}_{1:t-1}^{A_{t-1}^r}, \boldsymbol{s}_{1:t-1})\ f_\psi(\boldsymbol{\epsilon}_{1:t}^r \mid \boldsymbol{\epsilon}_{1:t-1}^{A_{t-1}^r})\, g_\psi(\boldsymbol{s}_{1:t} \mid \boldsymbol{\epsilon}_{1:t}^r)}{p_\psi(\boldsymbol{\epsilon}_{1:t-1}^{A_{t-1}^r}, \boldsymbol{s}_{1:t-1})\ q_\psi(\boldsymbol{\epsilon}_{1:t}^r \mid \boldsymbol{s}_{1:t}, \boldsymbol{\epsilon}_{1:t-1}^{A_{t-1}^r})} = \\ = \frac{f_\psi(\boldsymbol{\epsilon}_{1:t}^r \mid \boldsymbol{\epsilon}_{1:t-1}^{A_{t-1}^r})\, g_\psi(\boldsymbol{s}_{1:t} \mid \boldsymbol{\epsilon}_{1:t}^r)}{q_\psi(\boldsymbol{\epsilon}_{1:t}^r \mid \boldsymbol{s}_{1:t}, \boldsymbol{\epsilon}_{1:t-1}^{A_{t-1}^r})},\end{aligned}$$

where proposal importance density q_ψ is here selected equal to the prior and the law of $\boldsymbol{s}_{1:t} \mid \boldsymbol{\epsilon}_{1:t}$ is the Multinomial.
To guarantee that the weights are well-defined, they are then normalized.

The variant of sequential Monte Carlo method adopted in our problem is known as Conditional SMC, which leaves a fixed prespecified trajectory ($\boldsymbol{\epsilon}_{1:T}$,$B_{1:T}$) identical, whereas the resampling of the other $R-1$ particles is carried out as in standard SMC: this procedure ensures that $\boldsymbol{\epsilon}_{1:T}$ (and implicitly also its history $B_{1:T}$) survives all the resampling steps.

Here we present the outline of the sampler employed to update $\boldsymbol{\epsilon}_{1:T}$ given $\mathbf{s}_{1:T}$:

- Fix a history $B_{1:T} = (B_1, \dots, B_T)$
- At time $t = 1$:
  - for $r \neq B_1$, sample $\boldsymbol{\epsilon}_1^r$, with $\epsilon_{1j}^r \sim \mathcal{N}(0, 1)$;
  - compute and normalize the weights:

$$\omega_1(\boldsymbol{\epsilon}_1^r) \sim \text{Multinomial}(\boldsymbol{s}_1 \mid \boldsymbol{w}_1(\boldsymbol{\epsilon}_1^r))$$
$$\tilde{\omega}_1^r = \frac{\omega_1(\boldsymbol{\epsilon}_1^r)}{\sum_{r=1}^R \omega_1(\boldsymbol{\epsilon}_1^r)}.$$

- At time $t = 2, \ldots, T$:
  - choose the ancestor at time $t-1$ of each particle that needs to be generated at time t:
    for $r \neq B_t$, sample $A_{t-1}^r \sim \text{Discrete}(\tilde{\boldsymbol{\omega}}_{t-1})$;
  - for $r \neq B_t$, sample $\boldsymbol{\epsilon}_t^r \sim \prod_{j=1}^J \mathcal{N}(\epsilon_{tj}; \psi\epsilon_{t-1,j}^{A_{t-1}^r}, 1-\psi^2)$ and set $\boldsymbol{\epsilon}_{1:t}^r = (\boldsymbol{\epsilon}_{1:t-1}^{A_{t-1}^r}, \boldsymbol{\epsilon}_t^r)$;
  - compute and normalize the weights:

$$\omega_t(\boldsymbol{\epsilon}_{1:t}^r) \sim \text{Multinomial}(\boldsymbol{s}_t \mid \boldsymbol{w}_t(\boldsymbol{\epsilon}_t^r))$$
$$\tilde{\omega}_t^r = \frac{\omega_t(\boldsymbol{\epsilon}_{1:t}^r)}{\sum_{r=1}^R \omega_t(\boldsymbol{\epsilon}_{1:t}^r)}.$$

- Sample one particle r out of R particles with probability proportional to $\tilde{\omega}_T^r$: following its history $B_{1:T}^r = (B_1^r, \ldots, B_T^r = r)$, we obtain a sample of $\boldsymbol{\epsilon}_{1:T}$.

The update of $\boldsymbol{w}_{1:T}$ reduces to the computation of the stick-breaking weights given the sampled $\boldsymbol{\epsilon}_{1:T}$.

2.4. Cluster configuration

As last step we illustrate the final cluster estimates. We exploited the R package mcclust. The resulting partitions minimize the posterior expectation of a loss function on the space of partitions. There are two different functions to get point estimates for the clustering of the data: *Variation of Information (VI)* and *Binder*.
The results we will show further are the ones with VI loss function.

3. Clustering Problem

In the second part of the project we studied the clustering problem, in particular we focused on different clustering algorithms and we did a comparative study of cluster validity indices.

3.1. Standard Clustering Methods

Cluster analysis over our data corresponds to partition "male", "female" and "neutral" words at each decade.

Since the true partition is unknown we decided to compare different clustering algorithms.

3.1.1. K-Means

The first clustering method we analyzed is K-means.[5]

K-means clustering is the most commonly used unsupervised machine learning algorithm for partitioning a given data set into a set of k groups (i.e. k clusters), where k represents the number of groups fixed a priori. It classifies objects in clusters, such that the objects within the same cluster are as near as possible, whereas objects from different clusters are as far as possible. In particular, it tends to minimize at each step the *total within-cluster sum of square*:

$$WSS = \sum_{k=1}^{K} \sum_{x_i \in C_k} (x_i - \mu_k)^2$$

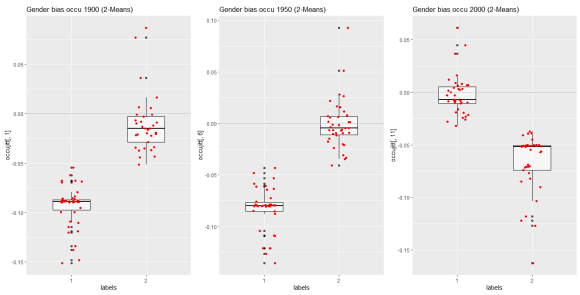
where μ_k are the mean points of the clusters, also called as *centroids*.

We also studied significant values which describe each cluster for each decade, such as centroid, mean value, min and max values, standard deviation and the number of objects.

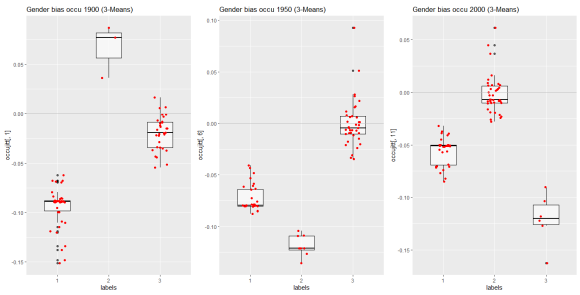
Fixed number of clusters

We considered the k-means clustering with numbers of clusters k from 2 to 5.

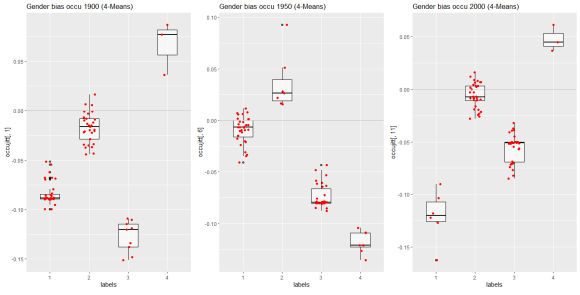
Occupations



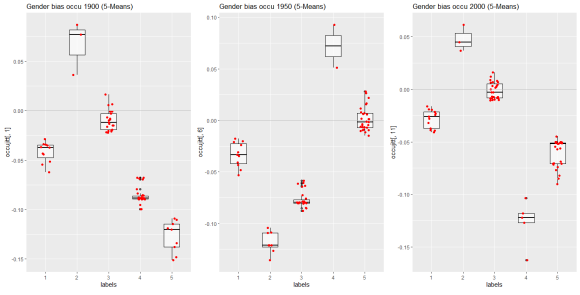
(a) K=2



(b) K=3



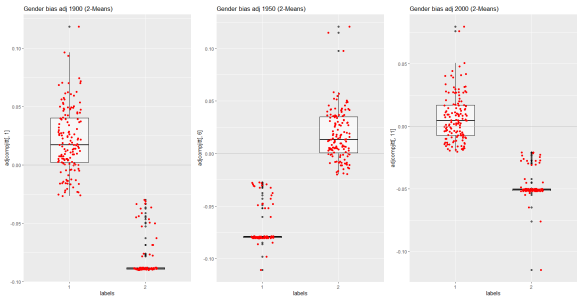
(c) K=4



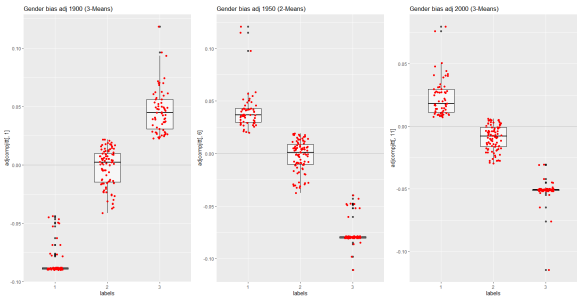
(d) K=5

[5] See "k_means_occu_2_5.xlsx" and "k_means_adj_2_5.xlsx"

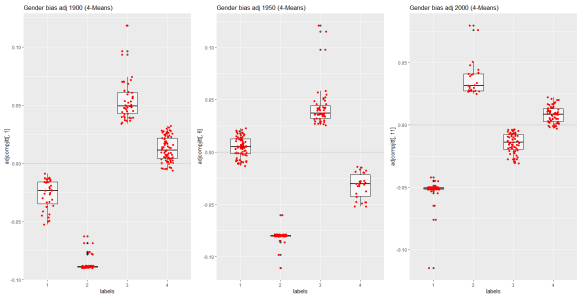
Adjectives



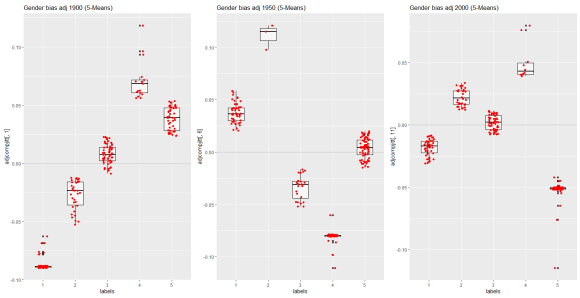
(a) K=2



(b) K=3



(c) K=4

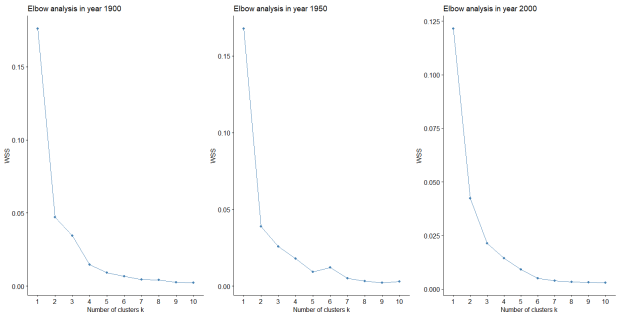


(d) K=5

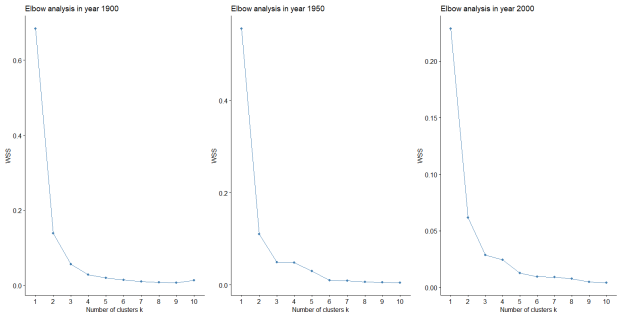
Since we are dealing with a dynamic clustering problem, where clustering configuration at time $t+1$ depends on the previous one at time t, we thought it was better to consider different values for k with respect to different decades.

Knee and elbow analysis

A simple method to choose the right number of expected clusters is the so called elbow analysis. The idea is to compute k-means with different number of clusters and then compute and plot the WSS for each k. The location of an elbow in this plot is generally considered as a good indicator of the appropriate number of clusters.



(a) Elbow Analysis for occupations



(b) Elbow Analysis for adjectives

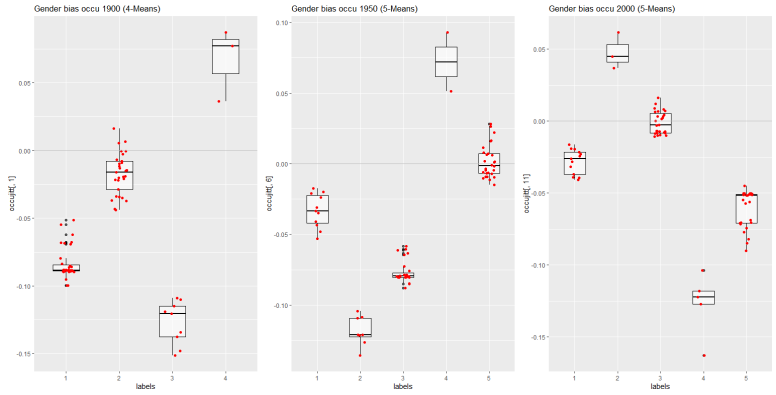
As a result we found the number of clusters for each decade according to the Elbow Analysis.

Results

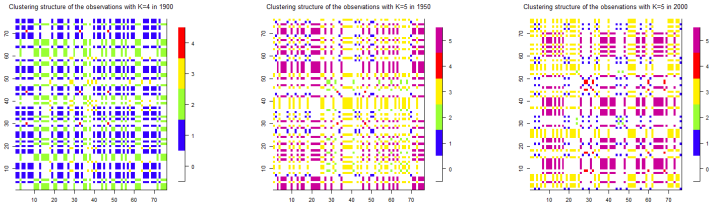
For what concerns occupation words, from the boxplots we can see that at the beginning of the century clusters are mainly male-oriented, while in the last decades there is a strong neutral-word cluster.

As a result of the elbow analysis plots it seems quite clear that we can consider k=4 or k=5 as the optimal number of clusters.

Occupations



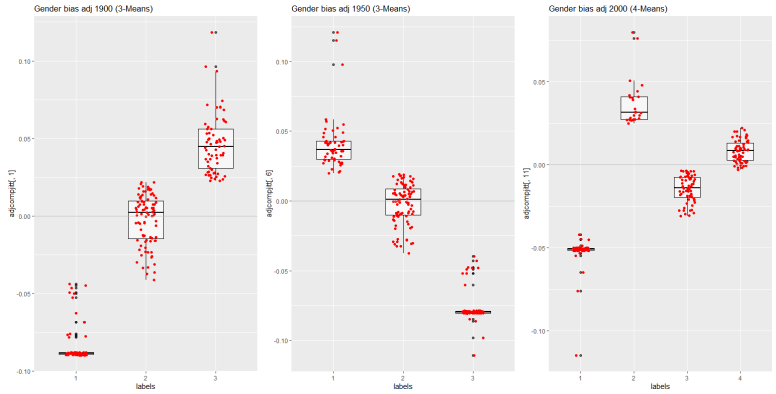
(a) K-means with Elbow Analysis for occupations



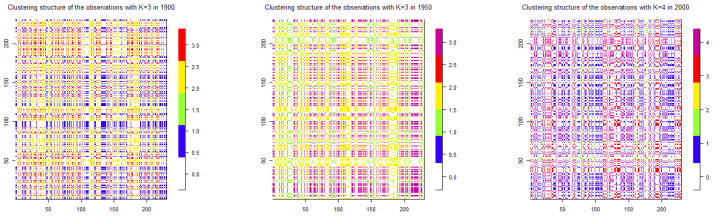
(b) Co-Clustering matrices for occupation

On the other hand according to adjective elbow analysis plots, we can set k=4 or k=3 as the optimal number of clusters.

Adjectives



(a) K-means with Elbow Analysis for adjectives



(b) Co-Clustering matrices for adjectives

Drawbacks of K-Means

From this first clustering analysis we can see that K-means method is really sensible to outliers, in fact we can notice that there are some clusters composed by only few samples and for this reason here are some meaningless groups.

Another critical point we faced is the fact that we need to choose the appropriate number of clusters in advance, which is not an easy task without prior knowledge of the data.

For these reasons we decided to use also PAM algorithm.

3.1.2. K-Medoids and PAM algorithm

The second clustering method we analyzed is k-medoid.[6]

The k-medoids algorithm is a clustering approach related to k-means clustering for partitioning a data set into k clusters.

The main difference from k-means corresponds to the definition of centroids for each cluster: instead of minimizing the distance from the object to a *centroid*, an imaginary point for k-means, this method involves the *medoids*, datapoints for which the average distance between them and all the other members of the cluster is minimal.

The main advantage with respect to k-means is the partitioning robustness, as a matter of fact the algorithm is less sensitive to noise and outliers. It happens because it uses medoids as cluster centers instead of means (used in k-means).

The most common k-medoids clustering methods is the PAM algorithm (Partitioning Around Medoids).

An useful approach to determine the optimal number of clusters is the silhouette method.

The Silhouette Method

The silhouette value is a measure of how similar an object is to its own cluster (cohesion) compared to other clusters (separation). An high value shows a good partition into clusters.

Since the main problem of clustering problem is the choice of the number of possible clusters, the idea used for PAM algorithm is to compute the silhouette coefficient for each k and then for each decade we take the one with the best silhouette index.

Results

PAM algorithm with best silhouette k (with k set from 2 to 10) produced the following results for occupations and adjectives.

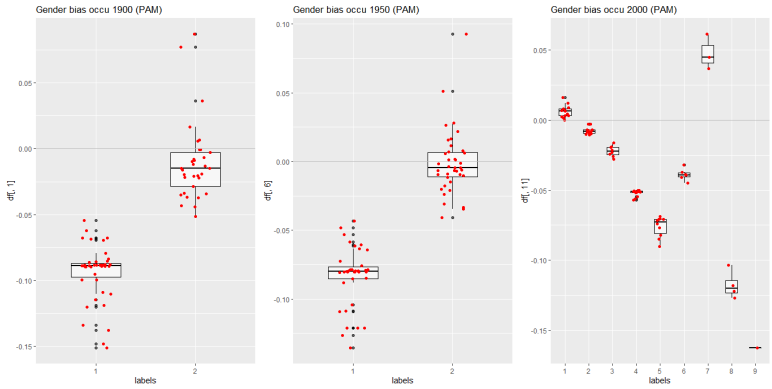
Occupations

For what concerns the occupation part, we found from the silhouette analysis that the optimal number of clusters can vary a lot among the decades:

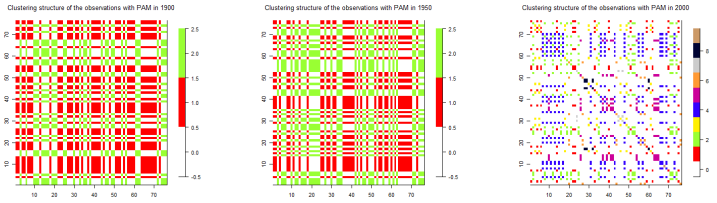
2 10 8 10 2 2 2 8 8 7 9

Note that a large k, such as 9 or 10, could be not very significant for a small dataset (Occupations has only 76 observations).

[6] See the code "pam_occu.R" and "pam_adj.R"



(a) K-means with Elbow Analysis for occupations

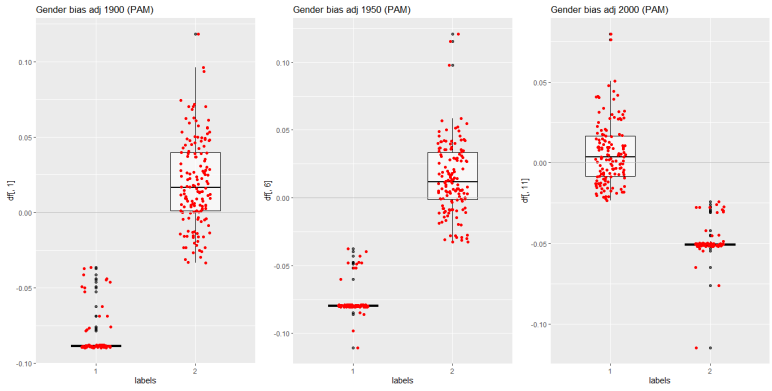


(b) Co-Clustering matrices for occupations

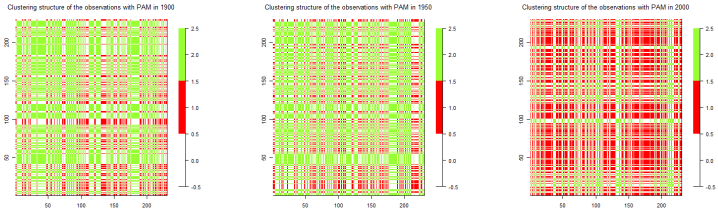
Adjectives

For what concerns the adjective part, we found from the silhouette analysis that the optimal number of clusters is more or less equal to 2 apart from the third and the eighth decades:

2 2 9 2 2 2 2 2 10 2 2



(a) K-means with Elbow Analysis for adjectives



(b) Co-Clustering matrices for adjectives

3.1.3. ARDP1

The last clustering method we used is AR1-DP, the one implemented by our code.

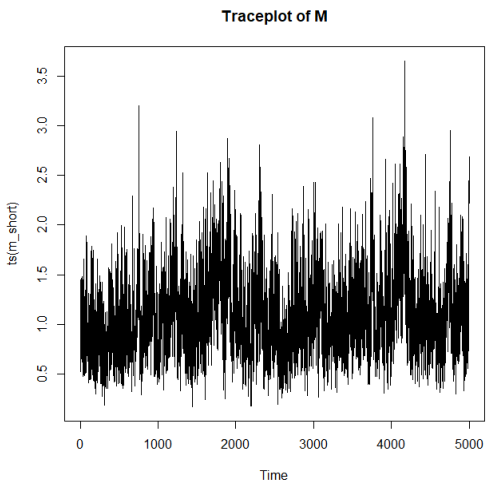
We decided to rely on this method because it describes in a good manner the dynamic clustering structure of our data.

Execution of the Code

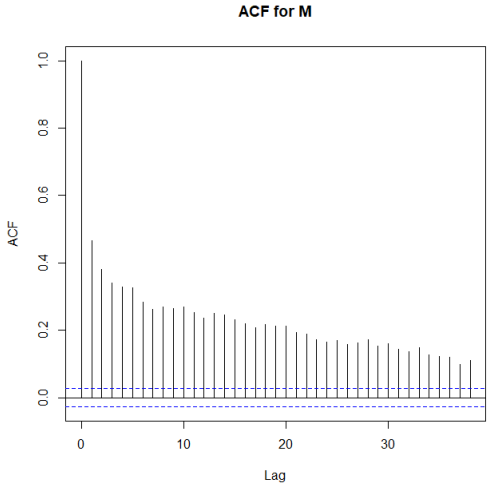
For what concerns the C++ code for the AR1-DP model we mainly focused our work on the occupations data since the adjectives dataset is almost four times larger and the Gibbs sampling strategy adopted is quite computanionally heavy. After comparing multiple runs we will showcase here the output of the code obtained choosing a burn-in period of 5000 iterations and a final sample size of 5000 with no thinning. Finally we chose to truncate the stick-breaking series of the DP to the N^{th} term, with N denoting the number of observations (76 for the occupations), as suggested in the paper by DeIorio et al.

Posterior analysis

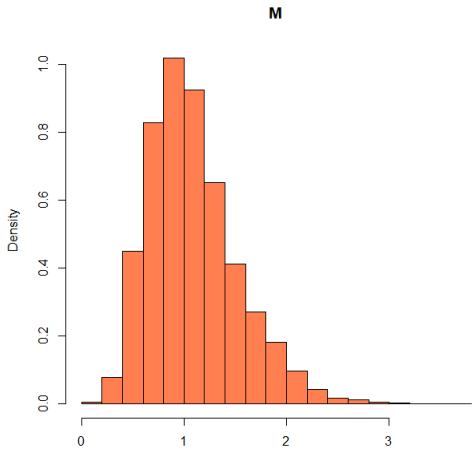
One of the parameters of the the model is M, the total mass parameter associated to the sequence of RPMs $(G_t)_{t\geqslant 1}$. In the pictures below we can see its traceplot, its autocorrelation function and its marginal posterior distribution. Although the traceplot is quite stable, the autocorrelation function shows a strong dependence among high order lags, thus indicating a slow mixing of the chain. To have a sample which looks more like an i.i.d sample from the marginal posterior one could try to consider a thinned chain. Nevertheless we kept all the 5000 points for our posterior analysis.



(a) Traceplot of M



(b) Autocorrelation function



(c) Marginal posterior distribution of M

Another important parameter in our model is ψ. It is particularly relevant because it is the parameter of the latent autoregressive process. As a consequence it is the parameter that determines the time dependency of the $(G_t)_{t\geqslant 1}$ and hence how the cluster estimates are related over time. First of all recall that the support of ψ is $(-1, 1)$. As highlighted in the picture below the marginal posterior of ψ puts most of its mass on $(0, 1)$ with a posterior mean of $\hat{\psi} = 0.491$. That shows somehow that there is a mild time dependence structure in the cluster estimates over the different decades.

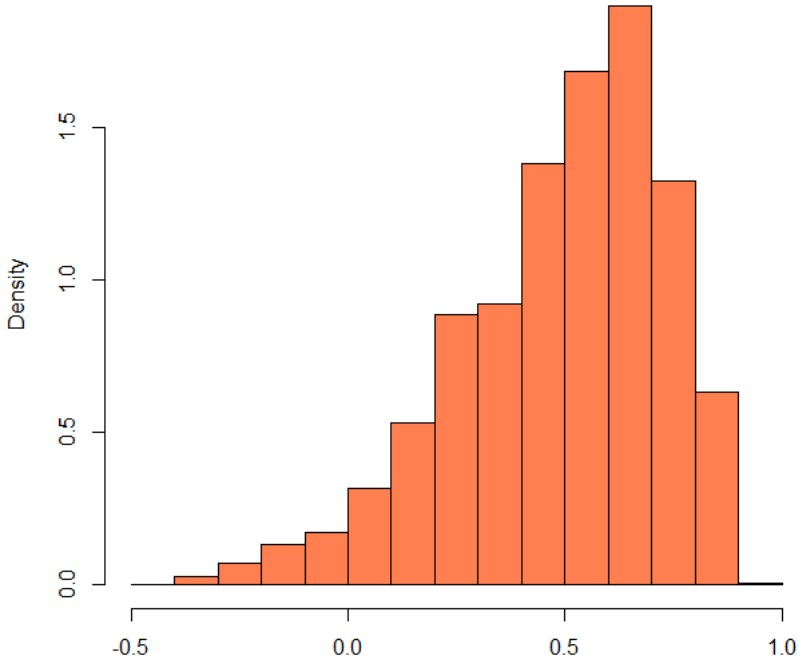


Figure 3.9: Marginal posterior distribution of ψ

As far as the number of clusters are concerned data slightly shift probability mass towards higher values. Recall that the the prior expected number of clusters K at time t is stationary and equal to 4. The posterior means of the number of clusters in the main decades, i.e in 1900,1950 and 200 are, respectively, $\hat{n}_{1900} = 6.036$, $\hat{n}_{1950} = 5.7381$, $\hat{n}_{2000} = 4.9998$. In the plot below it is displayed their marginal posterior distribution.

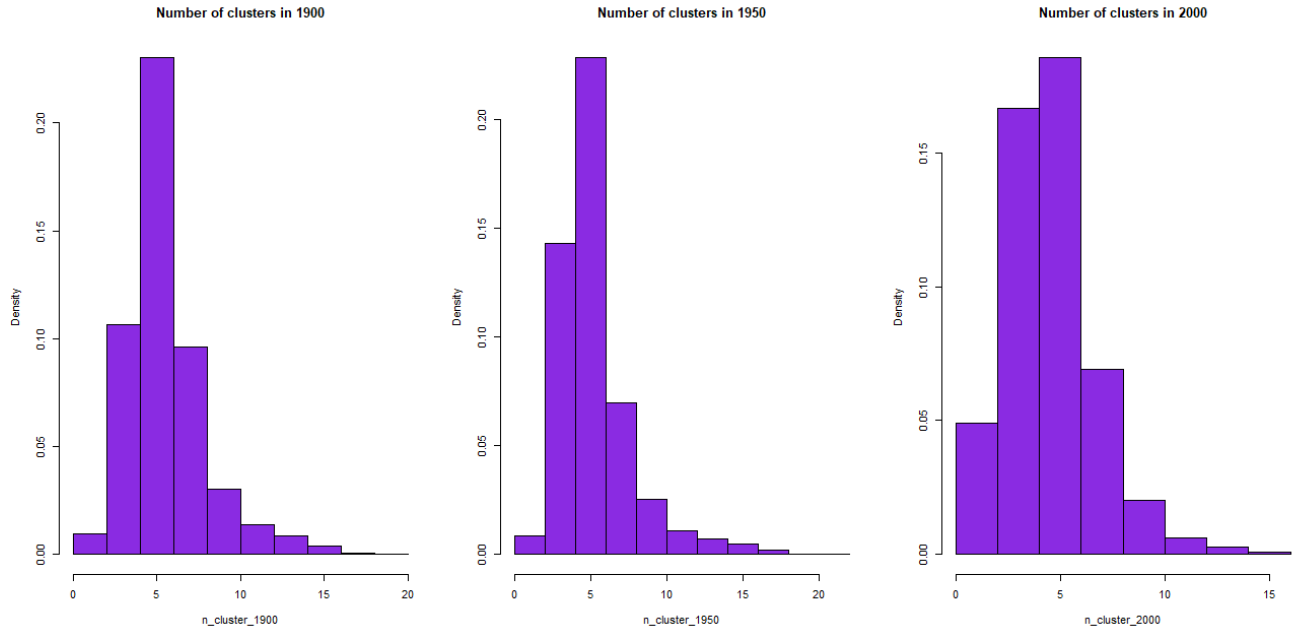


Figure 3.10: Marginal posterior distribution of the number of clusters in the main decades

When one works with a clustering algorithm they should be careful in judging the goodness of the cluster estimates. We can see that in some iterations of the chain our algorithm clusters data in a large number of clusters, specially in this case where we just have 76 data points. Most of the times when for an iteration of the MCMC we have a of clusters, most of them are actually singletons so they are not meaningful clusters. We will pay the same attention when we will try to recover a cluster estimate for our data starting from the posterior similarity matrices in the different decades. That is true in general: in a post-processing phase meaningless clusters should be merged into other clusters or should be considered as outliers, in some sense.

Cluster estimates

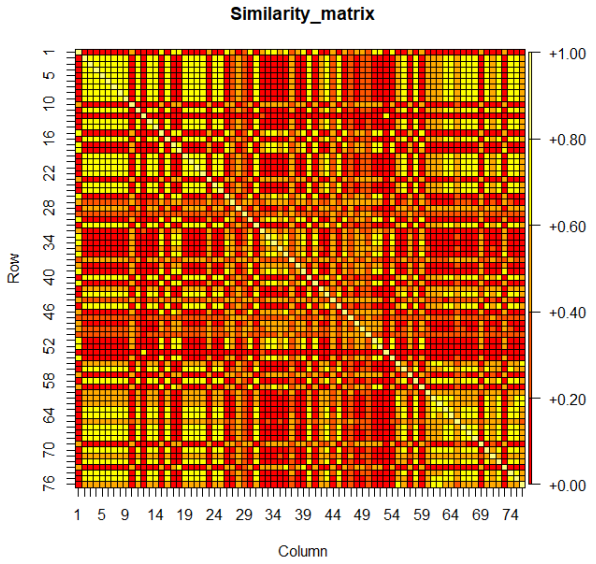
Our model is based on allocation variables s_{tj} for each latent variable θ_{tj}, as a consequence

$$s_{tj} = l \text{ if } \theta_{tj} = \theta_l$$

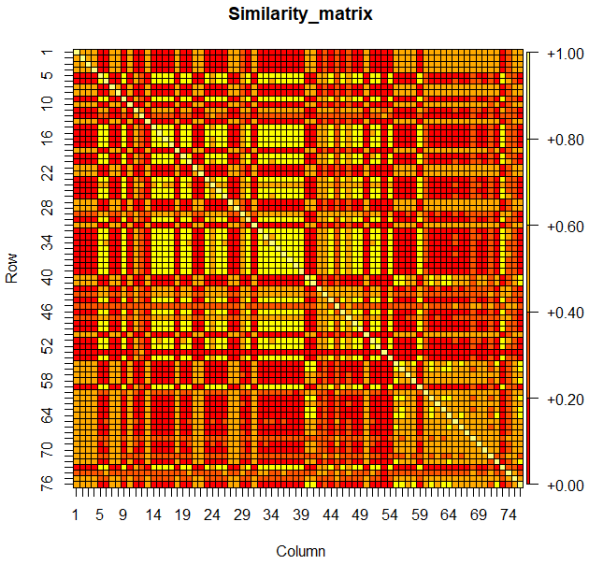
In this way we can introduce equivalence classes in our data, such that

$$Y_{tj} \sim Y_{th} \text{ if } \theta_{tj} = \theta_{th}$$

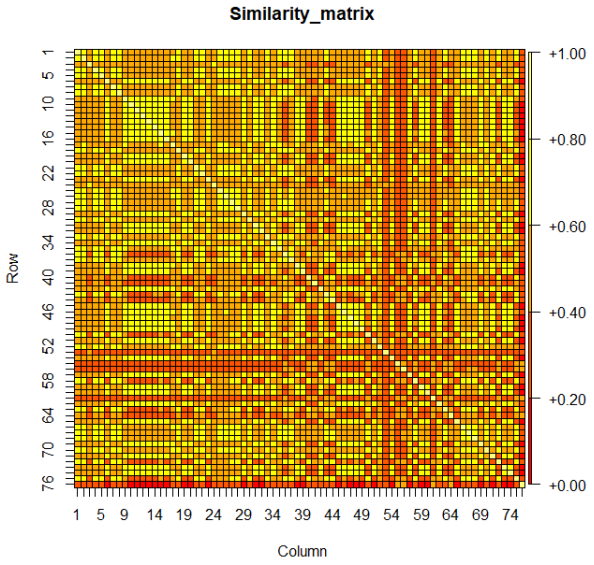
Notice how the information of the cluster estimates for each decade can be found in the MCMC sample of the matrix s, that is the matrix whose entries are the s_{tj} we have just recalled. We then would like to recover a cluster estimate from our MCMC sample. In order to do so we rely on the posterior similarity matrices. We fix a decade t and we define a $N \times N$ matrix whose entries $(i, j)_{i,j \in \{1,\dots,76\}}$ are the frequencies in the MCMC sample in which the observation i and the observation j are clustered together. In the plot below the posterior similarity matrices for the three main decades are showcased.



(a) Posterior similarity matrix for 1900



(b) Posterior similarity matrix for 1950



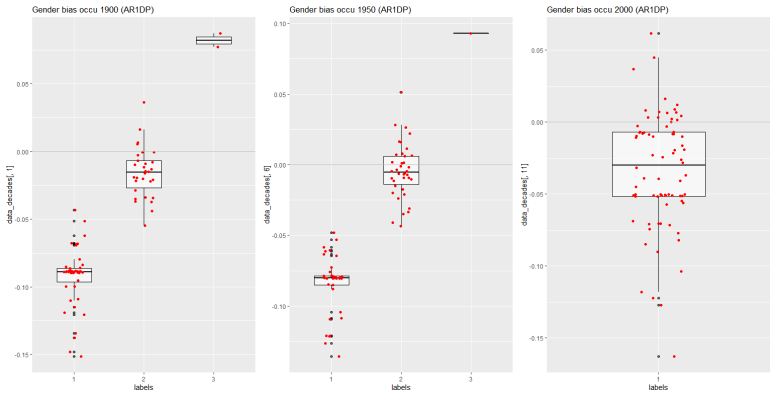
(c) Posterior similarity matrix for 2000

The posterior similarity matrices of these three decades are quite meaningful. In the one referring to 1900 the predominant colors are yellow and red while orange cells are more rare to find. That means that along the iterations of the chain the algorithm is consistent in the allocation structure of the data, that is: there are mostly couples of data points that are always clustered together (yellow cells) or couples that are never clustered together (red cells). This trend slightly changes in the fifties and completely turns around in the new millennium, where we can see from the posterior similarity matrix that the couples of data points that are never clustered together have almost completely vanished. That could mean that in the last decades the concept of occupations associated to a particular gender has failed or at least become weaker, which is quite reasonable if we think in comparison to the early 90's or even during WW2.

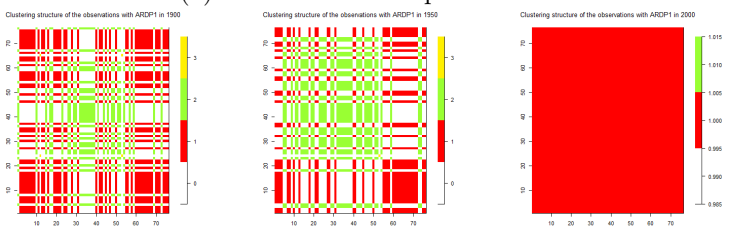
Posterior similarity matrices are a good summary from our MCMC sample but we would like to have a point estimate for the clustering of the date induced by our BNP mixture model. One

idea is to work in the space of square matrices looking for a matrix that, in binary representation, minimizes the distance from the posterior similarity matrix. This is equivalent to finding the partition of the data that minimizes the posterior expected loss, when the loss function is the Binder loss function. In this context, instead, we adopt a different loss function in the partition space that is the Variation of Information, as suggested in the paper by DeIorio et al.

From our analysis, the following result is achieved for occupations and adjectives.
Occupations



(a) AR1-DP for occupations



(b) Co-Clustering matrices for occupations

For what concerns occupation words, from the boxplots we can see that in the first decade of the last century data points are partitioned into 3 clusters that fairly describe the bias against women: the cluster containing occupations more closely associated to women consists just of the job "nurse" and "housekeeper" (in this sense it is more a couple of isolated points than a real cluster) the neutral cluster contains occupations such as "cashier", "teacher", and "musician". In the decade spanning from 1950 to 1959 just the occupation "nurse" is left in the female cluster, while there are still a lot of jobs strongly more closely associated to men, such as "statistician", "doctor" or "artist". For the first decade of the 21^{th} century all data points are actually clustered together. From this result we can guess the fact that during time the gender bias tends to vanish.

A very similar analysis can be made for the adjectives bias, analyzing the output of the same code but with the adjective dataset as input.

3.2. Cluster Analysis

Once we found a clustering partition, we wanted to understand if the proposed partition described our data in a good manner.

Since an optimal clustering algorithm does not exist and due to the fact that many clustering algorithms are not able to determine the number of natural clusters, we decided to rely on cluster validation analysis.
Cluster validation is a difficult task and lacks of theoretical background. However, the analysis of cluster validation techniques is a valid research for what concerns clustering algorithms' optimization.
We decided to rely on internal and external validation indices.

3.2.1. Internal Indices

Firstly we exploited internal indices. Internal validation indices are based on information intrinsic to the data and they evaluate the goodness of a clustering structure without external information. When the correct partition is not available it is possible to estimate the quality of a partition measuring how closely each instance is related to the cluster and how well-separated a cluster is from other clusters.
As a consequence, we would like the average distance within cluster to be as small as possible and the average distance between clusters to be as large as possible.
We can evaluate the goodness of a cluster by means of an internal validation index and then choose the one with the highest index as the best.
Our analysis involves two indices used indices for assessing the goodness of clustering: Dunn index and Silhouette index.

Notation for the next formulas: The dataset X has N points x_i; X is partitioned into K groups: C=$\{c_1, c_2, ..., c_K\}$; $d_e(x_i, x_j)$ is the euclidean distance between two points. I(C) is the index I computed for the partition C.

Dunn index

Dunn index is an internal clustering validation measure which focuses on the distance between each of the objects in the cluster and the objects in the other clusters.
If the dataset contains compact and well-separated clusters, the diameter of the clusters is expected to be small and the distance between the clusters is expected to be large.
Thus, Dunn index should be maximized.

$$D(C) = \frac{\min_{c_k \in C}\{\min_{c_l \in C - c_K}\{\delta(c_k, c_l)\}\}}{\max_{c_k \in C}\{\Delta(c_k)\}}$$

where: $\delta(c_k, c_l) = \min_{x_i \in c_k} \min_{x_j \in c_l} d_e(x_i, x_j)$ and $\Delta(c_k) = \max_{x_i, x_j \in c_k}\{d_e(x_i, x_j)\}$.

Silhouette index

The silhouette analysis measures how well an observation is clustered and it estimates the average distance between clusters. The silhouette plot displays a measure of how close each point in one cluster is to points in the neighboring clusters.
Thus, a large value of silhouette index (almost 1) means that data are very well clustered, while a small value (around 0) corresponds to the fact that datapoints can be separated into two clusters, otherwise datapoints with a negative value of this index are classified in a wrong

manner.

$$S(C) = \frac{1}{N} \sum_{c_k \in C} \sum_{x_i \in c_k} \frac{b(x_i, c_k) - a(x_i, c_k)}{\max\{a(x_i, c_k), b(x_i, c_k)\}}$$

where: $a(x_i, c_k) = 1/|c_k| \sum_{x_j \in c_k} d_e(x_i, x_j)$ and $b(x_i, c_k) = \min_{c_l \in C - c_k}\{1/|c_k| \sum_{x_j \in c_k} d_e(x_i, x_j)\}$.

3.2.2. Internal Indices: results for Occupations

At first we must say that both Dunn and Silhouette index are not well defined for single-cluster-partition; this is a problem, since the AR1DP estimates for the last three decades are indeed monocluster.

The silhouette indices show that our estimates are quite good: the performance is comparable with k-means, and not much worse than PAM (which has almost always the best silhouette, since it takes the number of cluster that optimize this index). See Figure 3.13.

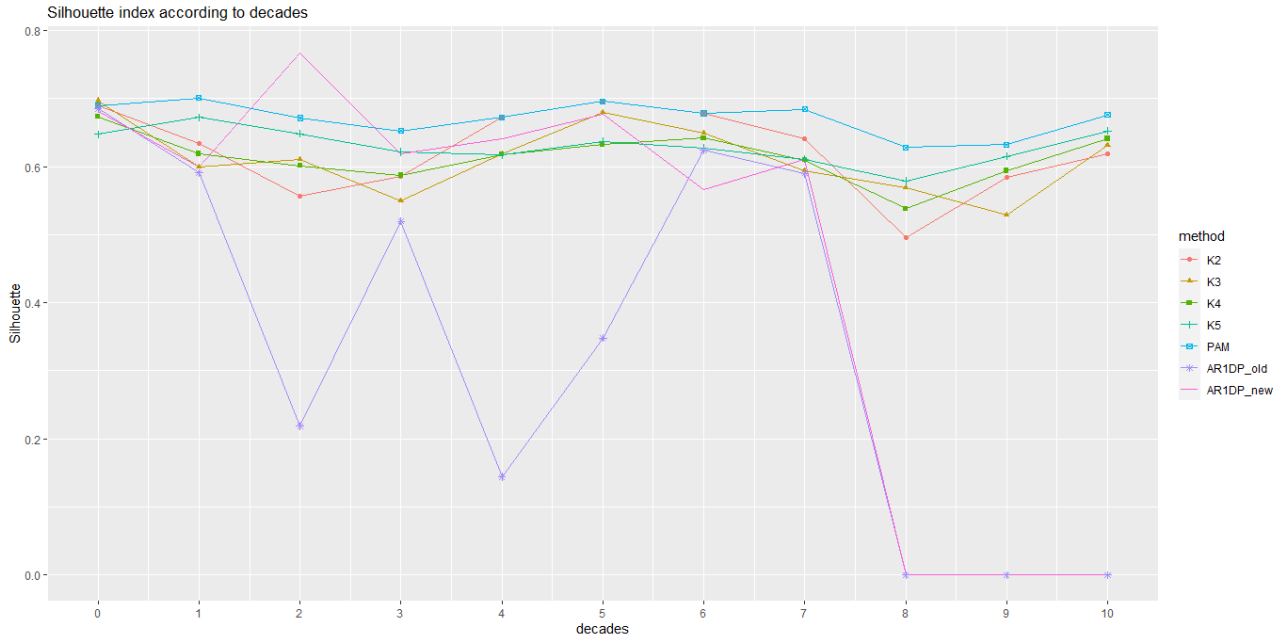


Figure 3.13: Silhouette indices. The last three decades for AR1DP methods are 0 by default.

The Dunn indices show apparently contradictory results: the fourth and the third decades have very high scores, whereas the other decades have quite poor scores. However, the third and the fourth decades are not very significant, since they might be considered monocluster (they both have two clusters, of which one has only one observation). See Figure 3.14

We can conclude that our cluster estimates are comparable with other methods; however we have to be careful since it easy to have a single cluster (or a very big cluster) within this dataset, and this can produce non significant internal indices.

3.2.3. External Indices

Secondly we exploited external indices. External validation indices measure the similarity between the output of the clustering algorithm and the correct partitioning of the dataset.

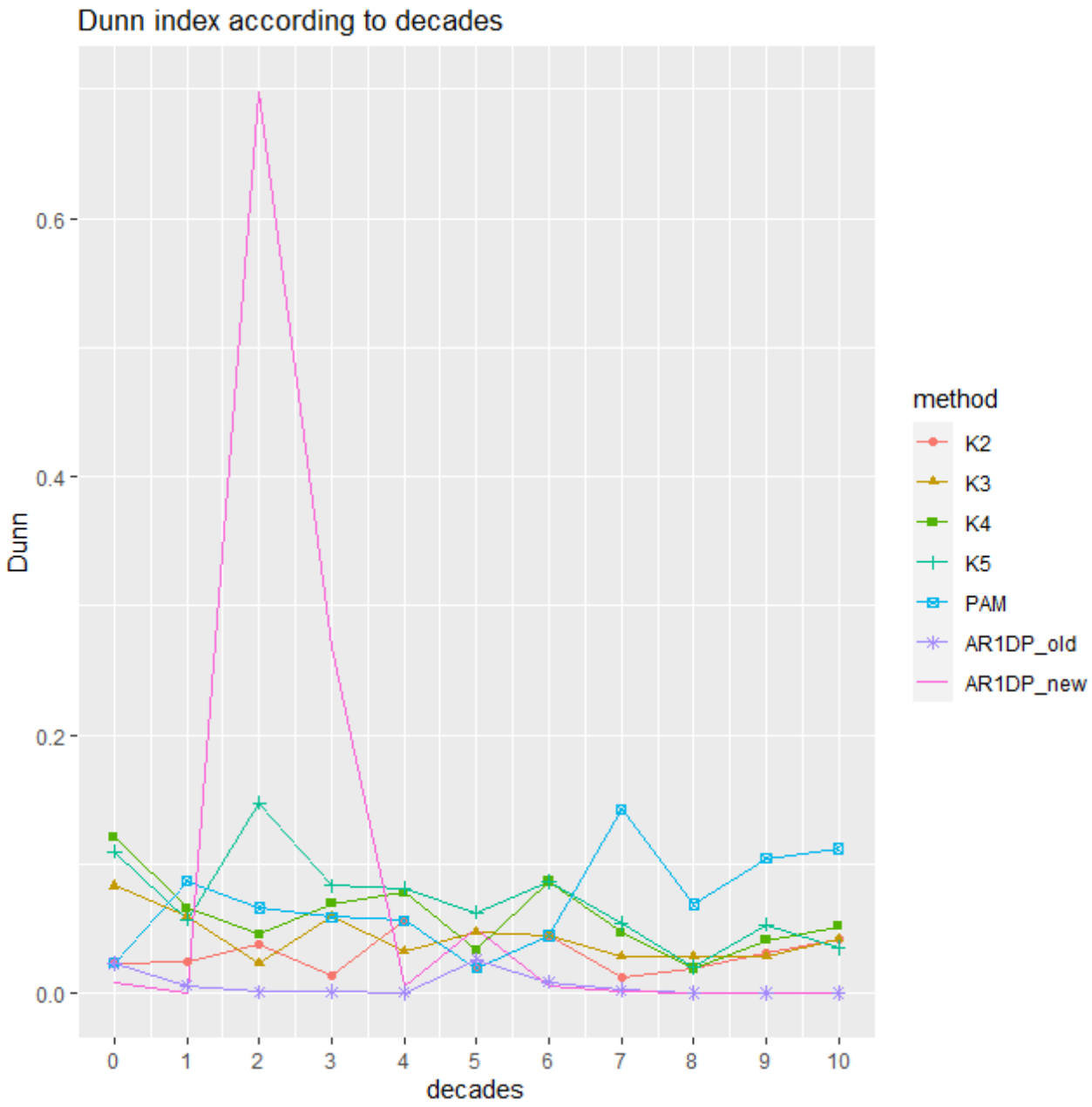


Figure 3.14: Dunn indices. The last three decades for AR1DP methods are 0 by default.

For our purpose, we exploited external indices to compare AR1DP results with PAM and K-means results since we do not have a correct cluster.
We considered The Jaccard, Fowlkes-Mallows and Adjusted Rand indices which belong to the same pair counting category, making them closely related. Some differences include the fact that they can exhibit biasing with respect to the number of clusters or the distribution of class sizes in a partition. All of them are a normalized measure of the similarity of two clustering output:

- if $I(C_A, C_B) \to 1$ the partitions C_A and C_B are almost identical;
- if $I(C_A, C_B) \to 0$ the partitions are very different from each other.

Notation for the next formulas
Consider the partition $U = \{u_1, u_2, ..., u_R\}$, where u_i, represents the cluster i and the partition $V = \{v_1, v_2, ..., v_C\}$, where v_j, represents the cluster j; n_{ij} is the number of objects belonging to u_i and v_j; $n_i = \sum_i n_{ij}$ and $n_j = \sum_j n_{ij}$

- a is the number of pairs of objects that are placed in the same group in both U and V:

$$a = \sum_{i,j} \binom{n_{ij}}{2}$$

- b is the number of pairs of objects belonging to the same group in U but different groups in V:

$$b = \sum_{i} \binom{n_i}{2} - \sum_{i,j} \binom{n_{ij}}{2}$$

- c is the number of pairs of objects belonging to the same group in V but different groups in U:

$$c = \sum_{j} \binom{n_j}{2} - \sum_{i,j} \binom{n_{ij}}{2}$$

Jaccard
The Jaccard coefficient measures similarities between different partitions into clusters. It also supports partitions with smaller number of clusters.

$$J = \frac{a}{a + b + c}$$

Adjusted rand index
The Rand Index measures similarities between different partitions into clusters. It tends to be indifferent to the number of clusters.

$$ARI = \frac{\sum_{i,j} \binom{n_{ij}}{2} - [\sum_i \binom{n_i}{2} \sum_j \binom{n_j}{2}]/\binom{n}{2}}{\frac{1}{2}[\sum_i \binom{n_i}{2} + \sum_j \binom{n_j}{2}] - [\sum_i \binom{n_i}{2} \sum_j \binom{n_j}{2}]/\binom{n}{2}}$$

Fowlkes-Mallows
The Fowlkes-Mallows Index measures similarities between different partitions into clusters. It supports partitions with smaller number of clusters.

One basic way to test the validity of this index is to compare two clustering partitions that are unrelated to each other. The index also shows similarity even when the noisy dataset has a different number of clusters than the clusters of the original dataset.
Thus making it a reliable tool for measuring similarity between two clusters.

$$FM = a\frac{\sqrt{a+b}\sqrt{a+c}}{(a+b)(a+c)}$$

3.2.4. External Indices: results for Occupations

We remember that, in our analysis, the external indices are not indicators for the goodness of the AR1DP method, but for the similarity with other methods.
The indices do not show a clear result: in some decades we can see good performance of AR1DP (high indices), whereas in others the indices are very low. Anyway, the three indices are quite consistent, namely there is not a decade in which the indices show completely different behaviours.
As for the internal indices, there are some decades which are not very significant because they have a monocluster structure.
See Figure 3.15 for 5-means vs. AR1DP and Figure 3.16 for PAM vs. AR1DP.

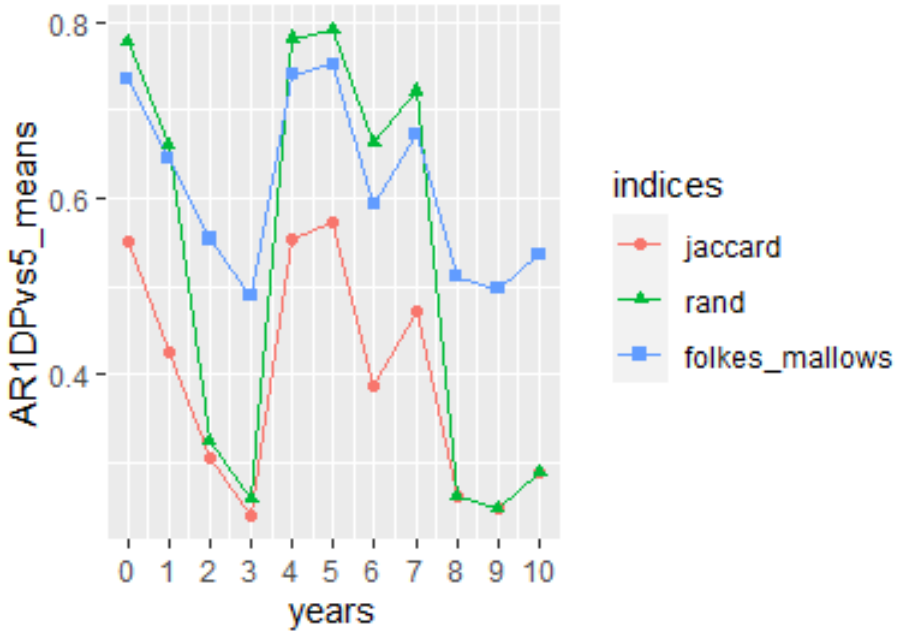


Figure 3.15: AR1DP vs 5-means external indices

As we expected, when one method generates a partition with a high number of clusters, Jaccard and Fowlkes-Mallows indices can be very low. That implies, for example, that some decades in AR1DP vs. PAM show a very poor performance

Figure 3.16: AR1DP vs PAM external indices

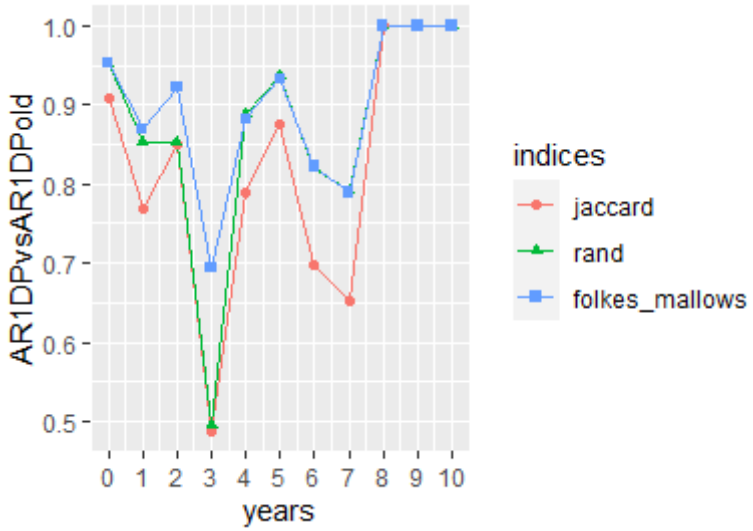


Figure 3.17: AR1DP vs AR1DP old external indices

Furthermore, we can see terrible results in the last three decades, where we got a single-cluster partition for the AR1DP method.

Lastly, we analysed the external validation for our results and the old AR1DP clustering: the indices show that the two estimates are quite similar for almost all the decades, with particularly high scores when the number of clusters is low in both methods.

4. The Code

The main aim of our project is to make more efficient (in C++ and Rcpp) a MCMC algorithm computing the posterior distribution in a Bayesian nonparametric dynamic mixture model (De Iorio, Guglielmi, Favaro, Ye, 2019) for clustering subjects over time.
We focused on two goals: make the code easier to be understood and make it a little computationally faster. We did not change the backbone of the project, since we did not find severe critical issues, although we have corrected a couple of small mistakes that were altering the output of the code.

4.1. Cleaning the code

We faced some issues we wanted to fix. Firstly original code functions were far more general in comparison to our purpose. For this reason we decided to delete unused functions to make the running execution less heavy.
Secondly another drawback we found was the nomenclature: the .cpp files, the functions and of the variables were definitely not intuitive and they did not suggest their uses. As a consequence we decided to change these names in order to make them also consistent with their use in the AR1DP workflow.[7]
Thirdly we wrote a lot of useful comments in order not to get lost while inspecting the tremendous number of operations written in this code.
Finally we modified some result outputs: some of them were made more readable from the user point of view, others were created in order to be used on R.

4.2. Optimizing the code

The main issue we found about the code was the so called pass-by-value. Pass by value happens when a called function makes a copy in memory of the actual parameter's value that is passed in. This is not a big issue when the argument is a primitive type, such as an integer number, while for heavier arguments pass by values becomes a point to focus on.
Since in the original code all the arguments were passed by value and due to the fact that we were dealing with a lot of heavier arguments, up to 4-levels arrays (matrices of matrices), we decided to pass all these demanding memory objects by const reference. In this manner objects are passed by their reference without creating a copy, improving memory and running efficiency.

It is not easy to quantify the amount of time we earned by this change, but we estimated that the code is approximately 10% faster.

[7] See the file OldNewFunctionsName.xlsx"

A. Dataset

Occupation words janitor, statistician, midwife, bailiff, auctioneer, photographer, geologist, shoemaker, athlete, cashier, dancer, housekeeper, accountant, physicist, gardener, dentist, weaver, blacksmith, psychologist, supervisor, mathematician, surveyor, tailor, designer, economist, mechanic, laborer, postmaster, broker, chemist, librarian, attendant, clerical, musician, porter, scientist, carpenter, sailor, instructor, sheriff, pilot, inspector, mason, baker, administrator, architect, collector, operator, surgeon, driver, painter, conductor, nurse, cook, engineer, retired, sales, lawyer, clergy, physician, farmer, clerk, manager, guard, artist, smith, official, police, doctor, professor, student, judge, teacher, author, secretary, soldier.

Adjective words headstrong, thankless, tactful, distrustful, quarrelsome, effeminate, fickle, talkative, dependable, resentful, sarcastic, unassuming, changeable, resourceful, persevering, forgiving, assertive, individualistic, vindictive, sophisticated, deceitful, impulsive, sociable, methodical, idealistic, thrifty, outgoing, intolerant, autocratic, conceited, inventive, dreamy, appreciative, forgetful, forceful, submissive, pessimistic, versatile, adaptable, reflective, inhibited, outspoken, quitting, unselfish, immature, painstaking, leisurely, infantile, sly, praising, cynical, irresponsible, arrogant, obliging, unkind, wary, greedy, obnoxious, irritable, discreet, frivolous, cowardly, rebellious, adventurous, enterprising, unscrupulous, poised, moody, unfriendly, optimistic, disorderly, peaceable, considerate, humorous, worrying, preoccupied, trusting, mischievous, robust, superstitious, noisy, tolerant, realistic, masculine, witty, informal, prejudiced, reckless, jolly, courageous, meek, stubborn, aloof, sentimental, complaining, unaffected, cooperative, unstable, feminine, timid, retiring, relaxed, imaginative, shrewd, conscientious, industrious, hasty, commonplace, lazy, gloomy, thoughtful, dignified, wholesome, affectionate, aggressive, awkward, energetic, tough, shy, queer, careless, restless, cautious, polished, tense, suspicious, dissatisfied, ingenious, fearful, daring, persistent, demanding, impatient, contented, selfish, rude, spontaneous, conventional, cheerful, enthusiastic, modest, ambitious, alert, defensive, mature, coarse, charming, clever, shallow, deliberate, stern, emotional, rigid, mild, cruel, artistic, hurried, sympathetic, dull, civilized, loyal, withdrawn, confident, indifferent, conservative, foolish, moderate, handsome, helpful, gentle, dominant, hostile, generous, reliable, sincere, precise, calm, healthy, attractive, progressive, confused, rational, stable, bitter, sensitive, initiative, loud, thorough, logical, intelligent, steady, formal, complicated, cool, curious, reserved, silent, honest, quick, friendly, efficient, pleasant, severe, peculiar, quiet, weak, anxious, nervous, warm, slow, dependent, wise, organized, affected, reasonable, capable, active, independent, patient, practical, serious, understanding, cold, responsible, simple, original, strong, determined, natural, kind.

Year 1900	Cluster 1 Mean(**-0.092400**) Std(0.022112) Min (-0.151266) Max (-0.043210) Count(44)	Cluster 2 Mean(**-0.015719**) Std(0.018708) Min (-0.054627) Max (0.036365) Count(30)	Cluster 3 Mean(**0.082002**) Std(0.007158) Min (0.076940) Max (0.087064) Count(2)
	accountant, administrator, artist, athlete, auctioneer, author, bailiff, clerk, collector, dancer, dentist, designer, driver, economist, engineer, farmer, geologist, guard, inspector, judge, lawyer, librarian, manager, mason, mathematician, midwife, official, photographer, physician, physicist, police, postmaster, professor, psychologist, retired, secretary, sheriff, shoemaker, smith, soldier, statistician,	architect, attendant, baker, blacksmith, broker, carpenter, cashier, chemist, clergy, clerical, conductor, cook, doctor, gardener, instructor, janitor, laborer, mechanic, musician, operator, painter, pilot, porter, sailor, sales, scientist, surgeon, tailor, teacher, weaver	housekeeper, nurse
Year 1950	Cluster 1 Mean(**-0.084316**) Std(0.020819) Min (-0.135664) Max (-0.048127) Count(37)	Cluster 2 Mean(**-0.004470**) Std(0.019599) Min (-0.043415) Max (0.051286) Count(38)	Cluster 3 Mean(**0.092850**) Std(0.000000) Min (0.092850) Max (0.092850) Count(1)
	accountant, administrator, artist, bailiff, blacksmith, cashier, clerk, doctor, driver, engineer, farmer, geologist, guard, janitor, judge, laborer, lawyer, librarian, manager, mathematician, midwife, official, physician, pilot, police, postmaster, professor, retired, sales, secretary, sheriff, shoemaker, smith, soldier, statistician, student, surveyor	architect, athlete, attendant, auctioneer, author, baker, broker, carpenter, chemist, clergy, clerical, collector, conductor, cook, dancer, dentist, designer, economist, gardener, housekeeper, inspector, instructor, mason, mechanic, musician, operator, painter, photographer, physicist, porter, psychologist, sailor, scientist, supervisor, surgeon, tailor, teacher,	nurse
Year 2000	Cluster 1 Mean(**-0.033814**) Std(0.040248) Min (-0.162792) Max (0.061585) Count(76)		
	accountant, administrator, architect, artist, athlete, attendant, auctioneer, author, bailiff, baker, blacksmith, broker, carpenter, cashier, chemist, clergy, clerical, clerk, collector, conductor, cook, dancer, dentist, designer, doctor, driver, economist, engineer, farmer, gardener, geologist, guard, housekeeper, inspector, instructor, janitor, judge, laborer, lawyer, librarian, manager, mason, mathematician, mechanic, midwife, musician, nurse, official, operator, painter, photographer, physician, physicist, pilot, police, porter, postmaster, professor, psychologist, retired, sailor, sales, scientist, secretary, sheriff, shoemaker, smith, soldier, statistician, student, supervisor, surgeon, surveyor, tailor, teacher, weaver		

Figure A.1: Occupations

B. Tables of validation analysis

SILHOUETTE	1900	1910	1920	1930	1940	1950	1960	1970	1980	1990	2000
K2	0.6897907	0.6342712	0.5561045	0.5852063	0.6726567	0.6960311	0.6782981	0.6412069	0.4952174	0.5848705	0.6193215
K3	0.6969573	0.5997780	0.6106737	0.5499030	0.6194335	0.6791568	0.6488454	0.5933896	0.5684752	0.5289302	0.6314071
K4	0.6733892	0.6189638	0.6014219	0.5868801	0.6178228	0.6320274	0.6420882	0.6093709	0.5381955	0.5939950	0.6411005
K5	0.6474549	0.6723042	0.6484332	0.6217810	0.6168911	0.6372144	0.6272662	0.6109813	0.5787980	0.6152885	0.6524048
PAM	0.6897907	0.7011158	0.6714296	0.6526421	0.6726567	0.6960311	0.6782981	0.6845660	0.6279822	0.6321520	0.6761700
AR1DP_old	0.6846027	0.5907697	0.2187852	0.5195879	0.1439982	0.3482570	0.6239349	0.5900329	0.0000000	0.0000000	0.0000000
AR1DP_new	0.6811470	0.5994983	0.7671293	0.6188533	0.6415914	0.6772997	0.5663750	0.6098644	0.0000000	0.0000000	0.0000000

DUNN	1900	1910	1920	1930	1940	1950	1960	1970	1980	1990	2000
K2	0.023405701	0.0242704605	0.0379174356	0.013361605	0.0564762815	0.01929934	0.044433102	0.011679047	0.01935053	0.03039540	0.04131265
K3	0.082929544	0.0595329637	0.0229905519	0.059860668	0.0323062016	0.04743164	0.044933156	0.028814236	0.02856352	0.02815394	0.04197870
K4	0.120682613	0.0655776871	0.0455522060	0.069145913	0.0776668904	0.03350630	0.086231486	0.047014642	0.01889619	0.04114437	0.05185489
K5	0.109621854	0.0570044840	0.1471853536	0.083338270	0.0814336412	0.06243054	0.086231486	0.053904997	0.02027926	0.05327815	0.03509107
PAM	0.023405701	0.0867523338	0.0659887565	0.058923169	0.0564762815	0.01929934	0.044433102	0.142630139	0.06863811	0.10409923	0.11174119
AR1DP_old	0.022870404	0.0057948720	0.0007630977	0.001282795	0.0002400412	0.02595294	0.008800464	0.002543268	0.00000000	0.00000000	0.00000000
AR1DP_new	0.008223156	0.0001849391	0.6977498559	0.267685672	0.0055525083	0.04975308	0.005707660	0.000769089	0.00000000	0.00000000	0.00000000

Figure B.1: Silhouette and Dunn indices

AR1DP vs 3-means	Jaccard	Rand	Folkes-Mallows	n clusters AR1DP
1900	0.88002771139144	0.93929827213287	0.93636488914485	3
1910	0.53494924306865	0.71052628755569	0.71464937925338	2
1920	0.43185904622077	0.44561403989791	0.65623271465301	2
1930	0.32642117142677	0.33894735574722	0.56680476665496	2
1940	0.74164342880249	0.86982458829875	0.85591191053390	3
1950	0.72795623540878	0.86035090684890	0.84548556804657	3
1960	0.55367910861969	0.74035090208050	0.71909856796264	3
1970	0.50189632177350	0.72350877523422	0.68404191732406	3
1980	0.41649124026298	0.41649124026298	0.64536130428314	1
1990	0.34280702471730	0.34280702471730	0.58549726009368	1
2000	0.42596492171287	0.42596492171287	0.65265989303588	1

AR1DP vs 4-means	Jaccard	Rand	Folkes-Mallows	n clusters AR1DP
1900	0.67496544122695	0.83508771657945	0.81410259008407	3
1910	0.42671927809715	0.64315789937973	0.62989836931228	2
1920	0.38306307792663	0.39929825067520	0.61892092227935	2
1930	0.31436800956720	0.33192983269691	0.56024593114852	2
1940	0.61843973398208	0.81122809648513	0.78083896636962	3
1950	0.72128850221633	0.86035090684890	0.84357082843780	3
1960	0.52684777975081	0.73719298839565	0.70543277263641	3
1970	0.56196159124374	0.76807016134262	0.73810160160064	3
1980	0.31719297170639	0.31719297170639	0.56319886445995	1
1990	0.32526317238807	0.32526317238807	0.57031846046447	1
2000	0.38912281394004	0.38912281394004	0.62379711866378	1

AR1DP vs 5-means	Jaccard	Rand	Folkes-Mallows	n clusters AR1DP
1900	0.54971593618395	0.77754384279251	0.73610872030258	3
1910	0.42476302385330	0.65929824113845	0.64522206783294	2
1920	0.30738738179206	0.32561403512954	0.55442529916763	2
1930	0.24018725752830	0.25964912772178	0.48953223228454	2
1940	0.55754107236860	0.78280699253082	0.74184852838516	3
1950	0.57204610109329	0.79157894849777	0.75261610746383	3
1960	0.38676846027374	0.66175436973571	0.59280014038085	3
1970	0.47168552875518	0.72175437211990	0.67300868034362	3
1980	0.26140350103378	0.26140350103378	0.51127636432647	1
1990	0.24701754748821	0.24701754748821	0.49700859189035	1
2000	0.28771930932998	0.28771930932998	0.53639471530914	1

AR1DP vs PAM	Jaccard	Rand	Folkes-Mallows	n clusters AR1DP	n clusters PAM
1900	0.82189118862151	0.90350878238678	0.90237653255462	3	2
1910	0.27365270256996	0.57438594102855	0.51984542608263	2	10
1920	0.20792792737484	0.22877192497253	0.45599114894866	2	7
1930	0.12972973287105	0.15263158082962	0.36018013954162	2	10
1940	0.92392808198928	0.96140348911285	0.96052366495132	3	2
1950	0.92307692766185	0.96105265617370	0.96008533239364	3	2
1960	0.66724842786780	0.79929822683334	0.80055046081543	3	2
1970	0.29965636134147	0.64245611429214	0.54143738746643	3	8
1980	0.16982455551624	0.16982455551624	0.41209775209426	1	8
1990	0.13859649002552	0.13859649002552	0.37228548526763	1	10
2000	0.14105263352394	0.14105263352394	0.37556973099708	1	9

Figure B.2: AR1DP vs. k-means (K=3,4,5) and PAM

AR1DP vs old	Jaccard	Rand	Folkes-Mallows	n clusters AR1DP	n clusters old
1900	0.90841245651245	0.95263159275054	0.95215302705764	3	2
1910	0.76872247457504	0.85263156890865	0.86976927518844	2	10
1920	0.84900903701780	0.85298246145248	0.92141687870025	2	7
1930	0.48772677779197	0.49473685026168	0.69383931159975	2	10
1940	0.7891606092453	0.88807016611095	0.88226866722106	3	2
1950	0.87508672475814	0.93684208393096	0.93346554040908	3	2
1960	0.69763314723966	0.82070177793502	0.82200902700424	3	2
1970	0.65120375156401	0.79157894849777	0.78900688886642	3	8
1980	1	1	1	1	8
1990	1	1	1	1	10
2000	1	1	1	1	9

k-means vs PAM	Jaccard	Rand	Folkes-Mallows	n clusters AR1DP
1900	0.53968256711955	0.76596492528915	0.73087304830551	2
1910	0.60000002384185	0.89403510093689	0.77068573236465	10
1920	0.66822427511215	0.90035086870193	0.81628787517547	7
1930	0.50218975543975	0.88035088775542	0.70096147060394	10
1940	0.57142859697341	0.78842103481292	0.75592893362045	2
1950	0.55174827575685	0.77508771419525	0.73797059059143	2
1960	0.47266435623168	0.73263156414032	0.67811918258667	2
1970	0.51492536067964	0.86315786838530	0.70422828197475	8
1980	0.50428396463394	0.85789471864700	0.68611419200897	8
1990	0.48915988206865	0.86771929264068	0.68457704782486	10
2000	0.48121210932731	0.84982454776763	0.69146561622615	9

Figure B.3: AR1DP vs. AR1DP old and 5-means vs. PAM

Bibliography

[1] Maria De Iorio, Stefano Favaro, Alessandra Guglielmi, Lifeng Ye.
Bayesian nonparametric temporal dynamic clustering via autoregressive Dirichlet priors (2019)

[2] Maria De Iorio, Stefano Favaro, Alessandra Guglielmi, Lifeng Ye.
Bayesian nonparametric mixture modeling for temporal dynamics of gender stereotypes (2020)

[3] Maria De Iorio, Stefano Favaro, Alessandra Guglielmi, Lifeng Ye.
Supplementary material to "Bayesian Nonparametric Mixture Modeling for Temporal Dynamics of Gender Stereotypes" (2020)

[4] Bas Kleijn, Aad van der Vaart, Harry van Zanten.
Nonparametric Bayesian Statistics; Part I: some classical results (2012)

[5] Christophe Andrieu, Arnaud Doucet, Roman Holenstein.
Particle Markov chain Monte Carlo methods (2010)

[6] Olatz Arbelaitz, Javier Muguerza, Jesùs M.Pérez.
An extensive comparative study of cluster validity indices (2018)

[7] Alboukadel Kassambara.
Practical Guide To Cluster Analysis in R (2017)

[8] Bernard Desgraupes, University Paris Ouest, Lab Modal'X.
Clustering Indices (2017)

[9] Mayra Z. Rodriguez, Cesar H. Comin, Dalcimar Casanova.
Clustering algorithms: A comparative approach (2019)

[10] Micheal D. Escobar, Mike West.
Bayesian Density Estimation and Inference Using Mixtures (1995)

[11] Peter Müller, Fernando A. Quintana.
Nonparametric Bayesian Data Analysis (2004)

[12] Radford M. Neal.
Markov Chain Sampling Methods for Dirichlet Process Mixture Models (2000)