POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Executive Summary of the Project

A Symmetric Prior for Multinomial Probit Model

Author: Erica Espinosa, Mattia Gentile, Davide Lo Piccolo,
Sophie Retif, Alessandro Sala, Alessio Tranchida

Tutor: Matteo Gianella, Alessandra Guglielmi

Course: Bayesian Statistics

Academic year: 2021-22

Date: February 2022

Abstract

In the Multinomial Probit Model framework, the choice of a base category is necessary to uniquely identify the parameters of the model. On the other hand, it has been shown that this choice can strongly affect the performance of the model itself. To solve this problem, a symmetric prior has been proposed, which introduces a new identification strategy and allows to build a Gibbs sampler. This algorithm permits to sample in an efficient way all the quantities involved in the model and obtain better performance with respect to the classical asymmetric approach.

1. Introduction

The Multinomial Probit Model (MNP) is a generalization of the probit model used in cases of multiple categories. MNP models are popular in studies involving discrete choice data as in marketing, politics, and transportation studies. Within this framework, the MNP needs to address the identification problem, i.e. a reference system needs to be established, which in our case concerns the location and scale.

In order to set the scale it has been proposed to fix the trace of the utility covariance matrix. Instead, to resolve location indeterminacy, parameters are typically identified by selecting a base category relative to which the choice parameters are defined. However, working with such a setting would affects the prior predictive choice probabilities, which in turn affects the posterior inference. In order to overcome this issue, a new prior structure has been proposed.

2. The classical Multinomial Probit model approach

It is assumed that each agent $i = 1, ..., n$ is choosing among p mutually exclusive alternatives, whose importance can be encoded by a vector of latent Gaussian utilities $W_i = \{w_{ij}\}$ of length p. The underlying assumption is that each agent selects the alternative Y_i that maximizes his utility:

$$Y_i = \text{argmax}_j \, w_{ij} \tag{1}$$

The vector of utilities is represented by the following:

$$W_i = X_i\beta + \varepsilon_i \tag{2}$$

where $\varepsilon_i \overset{\text{iid}}{\sim} \mathcal{N}(0, \Sigma)$ capture variations in taste across agents, β is the vector of regression parameters and X_i is a matrix of covariates with the following structure:

$$X_i \;= [I_p \quad I_p \otimes (x_i^d)^T \quad x_i^a] \tag{3}$$

where the p-dimensional identity matrix I_p plays the role of the intercept terms, $I_p \otimes (x_i^d)^T$ is a $p \times (p \cdot k_d)$ matrix resulting from the Kronecker matrix multiplication, given k_d the number of covariates that vary by decision maker (e.g. buyer's age, sex). Finally x_i^a is a $p \times k_a$ matrix whose columns contain the values of the k_a variables concerning the alternatives (e.g., product prices). In more detail:

$$w_{ij} = \eta_j + \left(x_i^d\right)^T \xi_j + x_{ij}^a \delta + \varepsilon_{ij}$$

with ξ_j a column vector of length k_d and δ a column vector of length k_a so that

$$\beta^T = (\eta_1, ..., \eta_p, {\xi_1}^T, ..., {\xi_p}^T, \delta^T)$$

making β a vector of length $p + (p \cdot k_d) + k_a$.

As previously reported, to identify the location it is needed to select a base category. Without loss of generality, the choice of the first one as base leads to the following transformation ${W_i}^* = T_{bc} W_i$ where

$$T_{bc} = [-J_{p-1} \quad I_{p-1}]$$

with J_{p-1} a column vector of ones with length $p-1$. In practice, this means to subtract the first utility from the others. In fact, for $j > 1$ this gives:

$$\begin{aligned} w_{ij}^* &= w_{ij} - w_{i1} \\ &= \eta_j + (x_i^d)^T \xi_j + x_{ij}^a \delta + \varepsilon_{ij} - (\eta_1 + (x_i^d)^T \xi_1 + x_{i1}^a \delta + \varepsilon_{i1}) \\ &= \eta_j - \eta_1 + (x_i^d)^T (\xi_j - \xi_1) + (x_{ij}^a - x_{i1}^a)\delta + (\varepsilon_{ij} - \varepsilon_{i1}) \end{aligned}$$

leading to $W_i^* = X_i^* \beta^* + \varepsilon_i^*$ where:

$${X_i}^* = [I_{p-1} \quad I_{p-1} \otimes (x_i^d)^T \quad T_{bc} x_i^a]$$

$$\beta^* = (\eta_2 - \eta_1, \ldots, \eta_p - \eta_1, (\xi_2 - \xi_1)^T, \ldots, (\xi_p - \xi_1)^T, \delta)$$

and $\varepsilon_i^* \overset{\text{iid}}{\sim} \mathcal{N}(0, \Sigma^* = T_{bc} \Sigma {T_{bc}}^T)$. Under this final reparametrization, the choice among the alternatives Y_i has the following form:

$$Y_i = \begin{cases} \text{argmax}_j w_{ij}^* + 1 & \text{if} \quad w_{ij}^* > 0 \\ 1 & \text{if} \quad \max_j w_{ij}^* < 0 \end{cases}$$

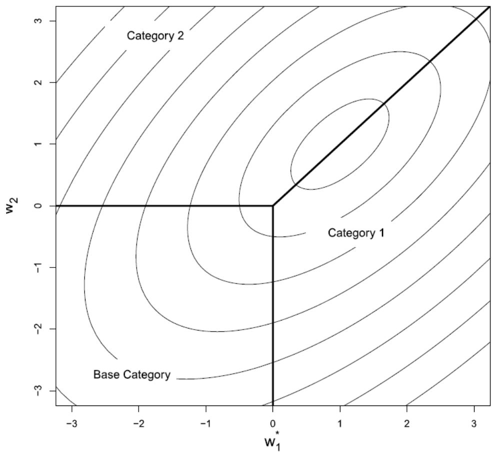

Figure 1: Multivariate normal contours corresponding to the base subtracted utility space for $p-1=2$. The base category standardization entails that the area in utility space allocated to the base category has a shape different from the regions allocated to non-base categories.

3. A symmetric prior for Multinomial Probit model

Although the choice of a base category should not influence inference, switching from one to another can result in substantial differences in posterior predictive probabilities. In fact, the base category standardization imposes an inherently asymmetric mapping from the utility space to probabilities. In Figure 1 it is shown an example with three categories and it is highlighted how the base category covers a region different both in shape and in probability from the other two.

As such, standard priors on Σ^* will generally correspond to asymmetric distributions over choice probabilities and, although it is expected that the impact of the prior would fade as the sample size increases, information in multinomial models update slowly relative to standard models of a continuous outcome, which means that asymmetries in the prior for an MNP model may persist in the posterior for sample sizes that are typical in business and economics applications.

In the paper *"A Symmetric Prior for Multinomial Probit Models"* [2], Burgette proposes a solution to overcome this dependence on the base category which pursues a prior that is invariant to relabelling of the outcome categories. In this new model, which we will refer to as symmetric MNP, to resolve location indeterminacy, a sum to zero constraint is imposed on latent Gaussian utilities, which leads to assume that the choice-specific covariates have zero mean.

Within this new setting, we require that a positive definite matrix of dimension $(p-1) \times (p-1)$ describes the covariance of all but one the dimensions of W_i. One of the main differences with respect to the classical approach is that, instead of restricting to a $p-1$ dimensional space, we still work in a p space but constraining ourselves on a $p-1$ dimensional hyperplane.

Moreover, the concept of *faux base category* is introduced. This category, which we will denote with b, represent the one left out at each iteration and, in contrast to previous MNP models, it yields the novelty of being learned according to Bayes rule, a more proper and robust procedure.

Thus, the new proposed model is the following:

$$
\begin{aligned}
b &\sim \text{unif}(\{1,\dots,p\}), && (4)\\
\Sigma_b &\sim p_{TR}(S_b, \nu_b), && (5)\\
R_b &= [\text{chol}(\Sigma_b)]^T, && (6)\\
R &= \begin{bmatrix} R_{1:(b-1)} \\ R_b^* \\ R_{b:p} \end{bmatrix}, && (7)\\
\beta_b &\sim \mathcal{N}(0,\ A), && (8)\\
\beta &= f(\beta_b), && (9)\\
W_i &\overset{\text{ind}}{\sim} \mathcal{N}(X_i\beta,\ RR^T), && (10)\\
Y_i &= \arg\max_j W_i && (11)
\end{aligned}
$$

where p_{TR} refers to the trace-restricted variant of the Imai and van Dyk prior [3], that is

$$p(\Sigma^*) \propto |\Sigma^*|^{-(\nu+p)/2}[\text{tr}(S\Sigma^{*-1})]^{-\nu(p-1)/2}\mathbb{1}_{\text{cond}}$$

and the condition is $\text{tr}(\Sigma_b) = p-1$. R_b is the transposed Cholesky decomposition of Σ_b such that $R_b R_b^T = \Sigma_b$. R^* is a row vector inserted into R_b at the b-th row such that the sum of each column of R is zero. In this formulation, β_b has dimension $(p-1)(k_d+1)+k_a$. The function f acts on β_b adding, in each sub-vector of length $p-1$ that corresponds to an agent-specific covariate (or the intercepts), an extra dimension in position b. These new inserted elements are chosen so that the resulting sub-vectors of dimension p sum to zero.

4. The algorithm

In order to estimate the model presented, we built a Gibbs sampler constructed on a transformed space:

$$(\alpha, \Sigma_b, b, W, \beta_b) \to (\alpha, \Sigma_b, b, \widetilde{W} = \alpha W,\ \widetilde{\beta}_b = \alpha\beta_b) \tag{12}$$

where α is a deterministic parameter computed in order keep $\text{tr}(\Sigma_b) = p-1$. This choice is taken in order to avoid mistakes pointed out in Jiao and van Dyk et al. [4].

The sampler proceeds in three steps:

- Draw $\widetilde{W} \mid Y, \widetilde{\beta}_b, b, \Sigma_b, \alpha$,
- Draw $\widetilde{\beta}_b \mid Y, \widetilde{W}, b, \Sigma_b, \alpha$,
- Draw $\alpha, \Sigma_b, b \mid Y, \widetilde{W}, \widetilde{\beta}_b$.

In particular, the code receives as input the matrix X, containing all the covariates as described previously, and the vector Y of the products chosen by each agent. It returns the estimated choice for each agent through MCMC method.

4.1. Initialization

Before entering in the details of the sampler we highlight the initialization we provided for all the quantities involved in the algorithm.

First of all, for each agent, the vector of latent utilities W_i is initialized by sampling from a normal distribution centered in zero and with covariance matrix the identity:

$$W_i \sim \mathcal{N}(0, I) \tag{13}$$

We then permute the elements in each W_i so that the respective maximum coincides with the observed Y_i.

We then initialize the *faux base category* b and α, the deterministic quantity used to transform our space; since their starting values are not impactful at all in the algorithm we can fix both of them to 1.
Going on, the matrix A (covariance of β_b) has to be defined as a symmetric positive definite matrix. Although choosing an identity matrix works fine, we opted, following McCulloch and Rossi et al. [5] insight, for an inverse wishart with $k + 4$ degrees of freedom and scale defined as a diagonal matrix of values $k + 4$ and dimension k, where $k = (p - 1)(k_d + 1) + k_a$:

$$A \sim \text{inv-Wishart}(k + 4, (k + 4)I) \tag{14}$$

Once we have defined matrix A, we can proceed by setting the starting point for the vector of the covariates' coefficients, β_b. This is sampled from a multivariate normal distribution of dimension $(p - 1)(k_d + 1) + k_a$ with mean zero and covariance matrix A:

$$\beta_b \sim \mathcal{N}(0, A) \tag{15}$$

After these steps we proceed by building the vector β in such a way that each subvector of length $p - 1$ corresponding to an agent-specific covariate (or the intercepts) receives an additional element in position b. This new element is chosen so that the sub-vector sums to zero.

Also the initialization of the matrix Σ_b is distributed as an inverse Wishart:

$$\Sigma_b \sim \text{inv-Wishart}(\nu_b, S_b) \tag{16}$$

Both the hyperparameters for this distribution may actually change based on the choice of the *faux base category* b but Burgette, Puelz and Hahn et al. [2] recommend selecting common values for them. In particular, according to Burgette, $\nu_b = p + 1$ and $S_b = (1 + c)I_{p-1} - cJ_{p-1}J_{p-1}^T$, where $c = \frac{1}{p-1}$ and J_{p-1} is a column vector of ones. These choices for the hyperparameters yield a good prior covariance structure and provide a sufficient level of regularization without being too informative.

To conclude this part we simply need to compute the new transformed quantities $\widetilde{W} = \alpha W$, $\widetilde{\beta}_b = \alpha\beta_b$. Now we are set up to discuss the procedure concerning the Gibbs sampler.

4.2. Full conditional of $\widetilde{W}$

In order to sample the matrix $\widetilde{W}$, we have to iterate one-by-one through all the elements of each $\widetilde{W}_{i,b}$. In practice, after dropping the b-th element of each column vector $\widetilde{W}_i$ and the corresponding elements in X_i and β, the full conditionals of elements $\widetilde{W}_{i,b}$ are truncated univariate normals. Moreover, we have three possible truncations based on the case we are in at each step.

Entering into details, we first removed from matrix X all the rows and columns related to category b. We did the same with matrix $\widetilde{W}$, removing row b but saving it in an auxiliary variable in order to plug it back in the matrix at the end to re-obtain a matrix of proper dimension. Again, we removed the b-th element from the vector $\widetilde{\beta}$. In this way, we obtained what we called, respectively, X_b, $\widetilde{W}_b$ and $\widetilde{\beta}_b$. We also computed $\widetilde{\Sigma}_b = \alpha^2\Sigma_b$ since it will be required later.

After all these preliminary steps, we can calculate means and variances for the truncated normals. To do so, in particular, we follow the same procedure explained by McCulloch and Rossi et al. [5]:

$$F = \widetilde{\Sigma}^{-1}_{(-j)(-j)}\sigma_{(-j)j} \tag{17}$$

$$\mu = x_{ij}\widetilde{\beta}_b + F^T(w_{i(-j)} - X_{i(-j)}\beta) \tag{18}$$

$$V = \sigma_{jj} - \sigma_{j(-j)}\widetilde{\Sigma}^{-1}_{(-j)(-j)}\sigma_{(-j)j} \tag{19}$$

where x_{ij} is the row vector corresponding to the j-th row of matrix X_i while $X_{i(-j)}$ is the submatrix created by deleting the j-th row from X_i, σ_{jj} is the element in position (j, j) of matrix $\widetilde{\Sigma}_b$, $\sigma_{j(-j)}$ and $\sigma_{(-j)j}$ are respectively the row and the column vector of $\widetilde{\Sigma}_b$ without j-th element and, finally, $\widetilde{\Sigma}_{(-j)(-j)}$ is the matrix $\widetilde{\Sigma}_b$ without both j-th row and j-th column.

Coming to the sampling from the truncated normal distributions for $\widetilde{w}_{i,b}$ we had to tackle the problem relative to how to obtain values in the right interval.
The strategy we employed is the one suggested by Albert and Chib et al. [1]: at each step, sample a proposal $\widetilde{w}_{ij} \sim \mathcal{N}(\mu, V)$ and, if the sampled value respects the conditions, we update the value in the matrix $\widetilde{W}_b$ otherwise, we reject the proposal and proceed to the next value.

The three possible truncations we will need to verify are the following:

- If $Y_i = j \neq b$: $\quad \widetilde{w}_{ij} > -0.5\sum_{k\notin\{j,b\}}\widetilde{w}_{ik} \quad$ and $\quad \widetilde{w}_{ij} > \max(\widetilde{w}_{ik} : k \notin \{j, b\})$,
- If $Y_i \neq b$ and $Y_i = k \neq j$: $\quad \widetilde{w}_{ij} < \widetilde{w}_{ik} \quad$ and $\quad \widetilde{w}_{ij} > -\sum_{l\notin\{j,b\}}\widetilde{w}_{il} - \widetilde{w}_{ik}$,
- If $Y_i = b$: $\quad \widetilde{w}_{ij} < \min\{-0.5\sum_{k\notin\{j,b\}}\widetilde{w}_{ik}, -(\max\{\widetilde{W}_{-\{j,b\}}\}) + \sum_{k\notin\{j,b\}}\widetilde{w}_{ik}\}$.

After having checked each value in matrix $\widetilde{W}_b$, we reintroduce the row vector corresponding to position b previously extracted in the same position and obtain in this way the updated $\widetilde{W}$.

Notice that in the original paper there is a slight mistake in these truncations which caused the entire algorithm to fail: we noticed that something was off and with the help of our tutor Matteo Gianella we were able to fix it.

4.3. Full conditional of $\widetilde{\beta}_b$

Regarding the transformed vector of coefficients $\widetilde{\beta}_b$, we draw it according to a multivariate normal distribution. This is a trivial consequence of the fact that we have a normal likelihood coupled with the normal prior we defined for β_b.

In the practice, we started by dropping from X and from $\widetilde{W}$ the elements corresponding to b in analogous fashion as previously. The peculiarity of this step is that all the agents are considered simultaneously since, as we can see in the following formulas, we have to compute summations over i (indexing the agents).

The updated formulas are:

$$\hat{\beta}_b = [\sum_{i=1}^{n} X_{i,b}^T \Sigma_b^{-1} X_{i,b} + A^{-1}]^{-1} [\sum_{i=1}^{n} X_{i,b}^T \Sigma_b^{-1} \widetilde{W}_{i,b}] \tag{20}$$

$$\widetilde{\beta}_b \sim \mathcal{N}(\hat{\beta}_b, \alpha^2 (\sum_{i=1}^{n} X_{i,b}^T \Sigma_b^{-1} X_{i,b} + A^{-1})^{-1}) \tag{21}$$

Last, in order to get back to proper dimensionality, we applied the same strategy used before and inserted into each subvector of length $p-1$ concerning agent-specific covariates (and intercept) an additional element in position b such that the new subvector sums to zero.

4.4. Full conditionals of α, Σ_b and b

This sampling is structured in few steps: first of all we update the *faux base category* b, then, we draw the intermediate quantity $\widetilde{\Sigma}_b$ and compute α and finally compute Σ_b from α and $\widetilde{\Sigma}_b$.
In detail, b is sampled from a multinomial distribution where the probability that each category j will be the next faux base category chosen by the algorithm is proportional to:

$$p(b \,|\, \widetilde{\beta}, \widetilde{W}) \propto |S_b + \sum_i (\widetilde{W}_{i,j} - X_{i,j}\widetilde{\beta}_j)(\widetilde{W}_{i,j} - X_{i,j}\widetilde{\beta}_j)^T|^{-(n+\nu_b)/2} \tag{22}$$

where $\widetilde{W}_{i,j}$, $X_{i,j}$ and $\widetilde{\beta}_j$ uses the same notation as in previous steps.

In this framework we had to face the problem that the previous quantity grows exponentially with the number of agents involved, eventually causing computational issues. So we decided, and we recommend this specific practice, to deal with log-probabilities and then normalize them through a softmax. In this way we managed to take care of the possible problems connected to overflow. Defined the probabilities in this way, we sampled from a multivariate distribution and updated b with the new drawn value.

Having updated the value of b, we compute again the various quantities X_b, $\widetilde{W}_b$ and $\widetilde{\beta}_b$ and with them sample $\widetilde{\Sigma}_b$ from an inverse Wishart. In practice:

$$\widetilde{\Sigma}_b \sim \text{inv-Wishart}(n + \nu_b, S_b + S) \tag{23}$$

where

$$S = \sum_i (\widetilde{W}_{i,b} - X_{i,b}\widetilde{\beta}_b)(\widetilde{W}_{i,b} - X_{i,b}\widetilde{\beta}_b)^T.$$

We are left to compute α and Σ_b:

$$\alpha = \sqrt{\frac{\text{tr}(\widetilde{\Sigma}_b)}{p-1}} \tag{24}$$

$$\Sigma_b = \widetilde{\Sigma}_b / \alpha^2 \tag{25}$$

In this way, we make sure that the constraint $\text{tr}(\Sigma_b) = p - 1$ is met at each iteration.

Last, we simply compute $W = \widetilde{W}/\alpha$ and $\beta = \widetilde{\beta}/\alpha$ in order to go back to the original space.

This concludes the algorithm.

5. Simulated datasets

We wanted to test our Gibbs sampler and to compare the results obtained with the one attained by the classical MNP model using each of the possible base categories. In order to do so we built different simulated dataset loosely based on some real dataset.

The multinomial probit model is often used to analyze the discrete choices of individuals recorded in survey data. Examples where the multinomial probit model may be useful, include the analysis of product choice by consumers in market research and the analysis of candidate or party choice by voters in electoral studies.
In particular, Imai and van Dyk et al. [3] apply their methods to a consumer choice model of clothing detergent purchases and margarine purchases that are available respectively in the MNP package and in the bayesm package in R.

We focused on the second dataset which is related to the purchase of 10 different types of margarine from 516 households. Multiple purchases for each individual are recorded so that there is a total amount of 4470 purchases. The dataset also contains demographic information about the households and in particular we chose to take into account the variable *income* expressed in U.S. dollars and the variable *family size* expressed through two dummy variables, the first indicating a size of 3-4 members and the second one ≥ 5 members.

We decided to build 34 simulated datasets based on the previously described dataframe. For each dataset we assumed that $n = 100$ consumers were choosing among the $p = 10$ products described in the `margarine` dataset and for each individual we took its income and family size reported in the real data.

To make the *income* data more evenly distributed, we applied a log-transformation to it. To make it differs from one simulated dataset to the other we added a random uniform component drawn in the interval $[-10, 10]$ to the original value, while its β-coefficient was computed in order to get correlation with the mean prices close to 0.9 so that individuals with higher income are more inclined to buy more expensive products. Then, the variable *family size* was generated randomly sampling with probability $(0.45, 0.45, 0.1)$ the size $(1-2, 3-4, \geq 5)$ and their coefficients were sampled from a uniform distribution in the interval $(0, 1)$. The covariate product price was generated considering the real data plus an uniform error that is randomly extracted from the interval $[-0.1, 0.1]$. Its coefficient was then drawn from $[-1.25, -0.75]$ so that if a product is relatively less expensive, it will be more popular. Regarding the simulated product-specific intercepts, they were built in order to have correlation equal to 0.9 with the mean prices so that more desirable products are more expensive, as one would expect.

Finally, once we generated our simulated datasets, for each agent we sampled the utility matrices W_i from

$$W_i \sim \mathcal{N}(X_i\beta, \Sigma),$$

where

$$\Sigma \sim \text{inv-Wishart}_{50}(I)$$

is a $p \times p$ matrix drawn from an inv-Wishart distribution with 50 degrees of freedom and mean the identity matrix I of dimension $50-p-1$. Then, the simulated vector of choices made by the consumers are computed taking the product that has higher utility.

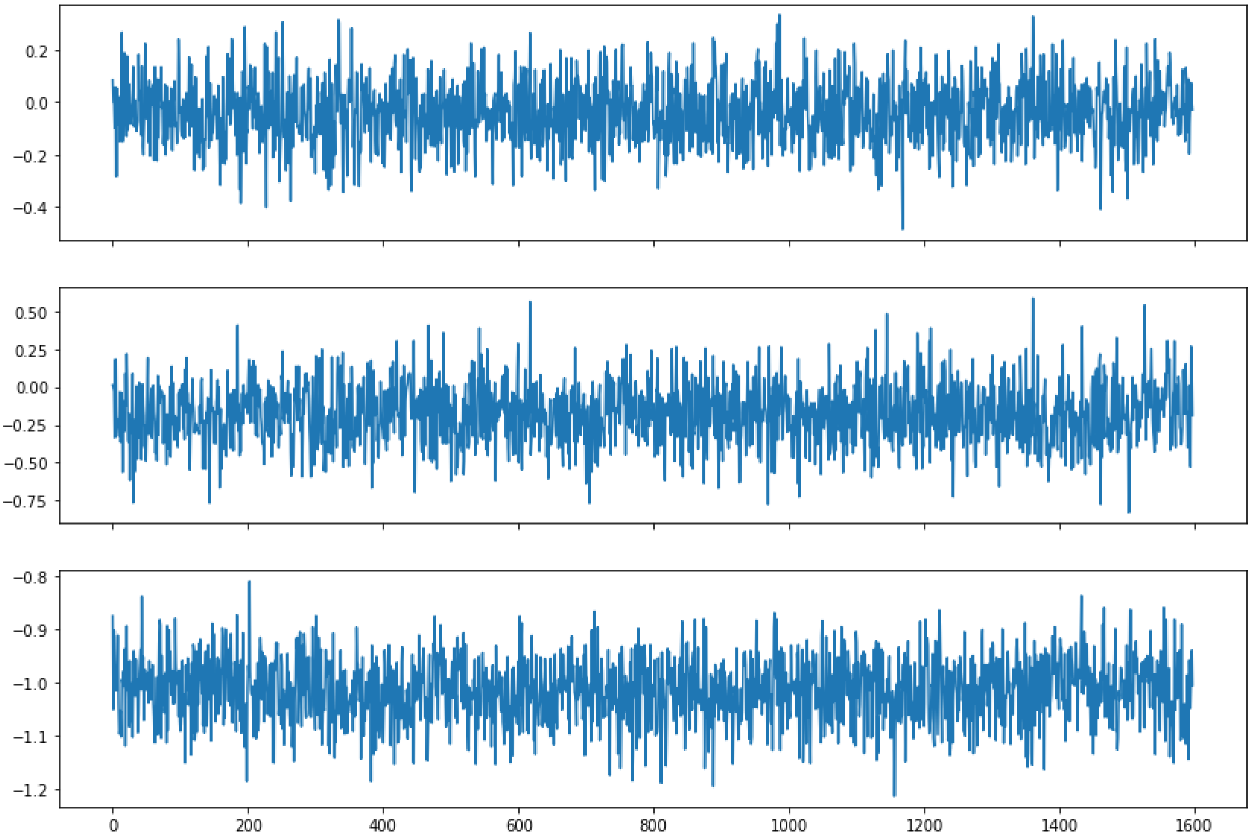

Figure 2: Traceplot of 10,000 iterations of the MCMC sampler with a burn-in of 2,000 and a thinning of 5 for three components of the β vector.

6. Results

As a matter of fact, we tested our Gibbs sampler on these very simulated datasets and compared it with the state of the art. Moreover, we looked at some diagnostic to assess the healthiness of our code.

Indeed, concerning this second task, we drew the traceplots regarding the MCMC behaviour of some variables sampled with our Gibbs sampler. In particular, we focused our attention on the components of β since, for instance, the elements of the covariance matrix Σ are not very informative and the individual utilities are way too many to be analyzed efficiently (they are equal to the number of products times the number of agents in the dataset).

In Figure 2 are represented the traceplots of three of these components, taken with a thinning of 5 over 8 thousand iterations after burn-in; as one can see, their behaviour is optimal: their value keeps jumping around the average value which instead remains basically constant, showing the typical *"fat hairy caterpillar"* shape and proving that the chain has come to a good mixing.

After that, we looked into the autocorrelation problem, again for the components of β, to see if there was any issue and, in particular, we used the `arviz` package for Python to generate autocorrelation plots. The results, of which we have an example in Figure 3, showed that a thinning was necessary but, after that, we had beautiful plots showing almost no correlation between consecutive draws, another proof of the good mixing of the chains.

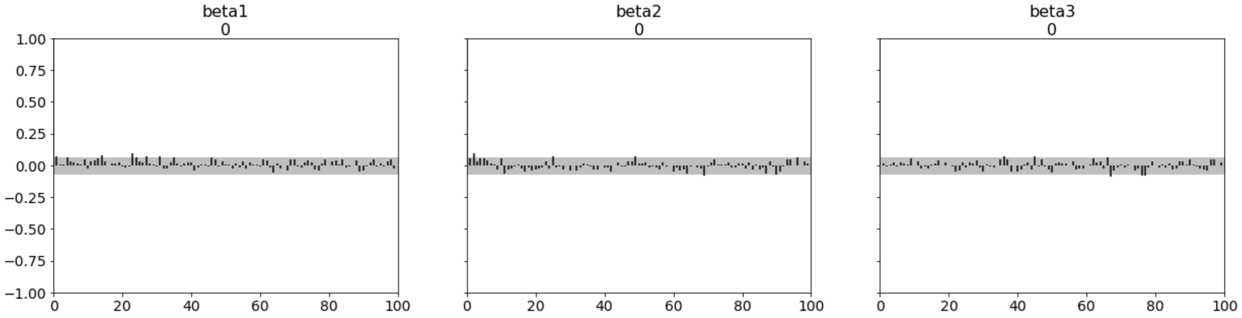

Figure 3: Autocorrelation plot of the chains of three components of the vector β computed with a thinning of 5.

Finally, we came to a direct comparison with the present state of the art, represented in this case by the mnp function in the homonymous R package [6]. Regarding the sMNP, the prediction has been done directly from the posterior draws of the utilities W: after the burn-in iterations, we considered, for each agent, which product they chose most often supposing that, at each iteration, they choose the one corresponding to the argmax of their vector of latent utilities. The MNP package, instead, offers a built-in predict function which we used on the same agents considering all possible base categories. Notice that we could only compare the results but we could not compare directly the processing speed of the two methods since our code has been developed in python with little to no regard for time optimization while the mnp function is coded in R and has been improved during the course of the last 15 years.

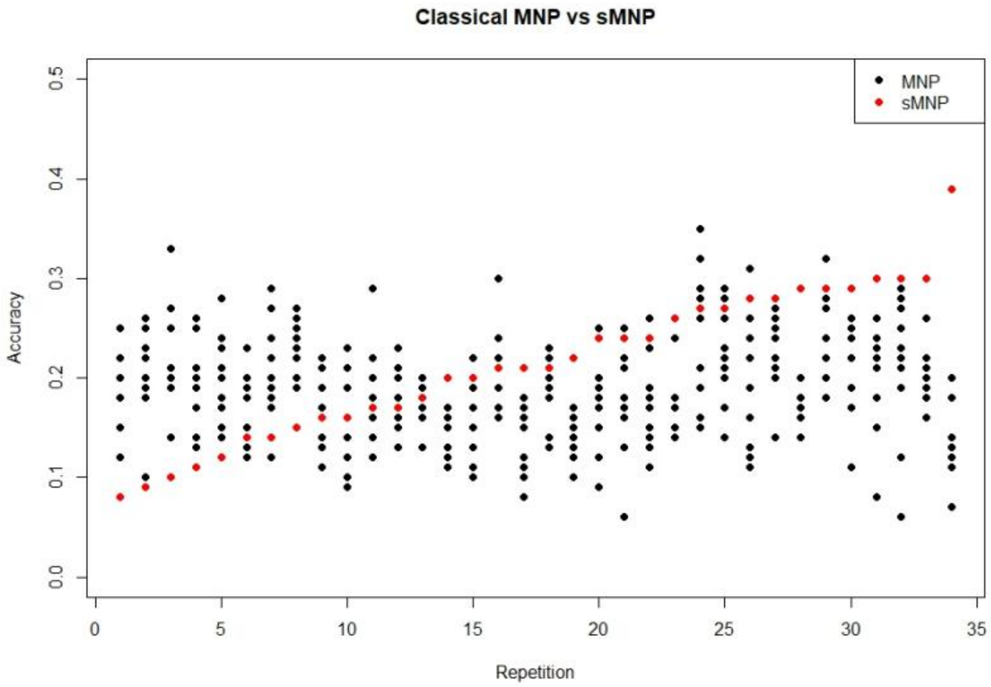

Figure 4: Accuracy of the models computed with MNP (in black), changing the base category, and with the sMNP (in red) of our 34 simulated datasets. It is notable how in 6 cases the sMNP works worse than any case of the MNP, in 11 cases it works better than any case of the MNP, and in the most of the remaining cases it works better than the mean of the others.

The results of the comparison can be seen in Figure 4 where the accuracy for each of the 34 datasets of the two models is shown. Values for the MNP model are represented in black and are calculated for each base category, while values for the sMNP are in red. It can be seen that there are cases where the accuracy of the sMNP are lower than all MNP models, but, in the majority of cases, sMNP outperforms the average results of the MNP models.

Therefore, in our case, it can be said that sMNP on average performs better than MNP and thus represents a more reliable procedure.

7. Conclusions

In the Bayesian MNP, appropriately handling the prior is necessary in order to obtain reliable predictions. The conducted analysis show that, seen the variability of results obtained with MNP, the willingness of obtaining smoother results is more preferable. In fact, the output of the classical model can be very sensible with respect to the choice of the base category while with the sMNP it appears evident that the predictions keep a robust tendency.

Embracing the sMNP setting it happens that the environment of the analysis is not formally identified since we impose a condition on the trace instead of comparing directly to one of the categories. However, little is lost in this case: firstly, the model is identified at each iteration conditionally on the faux base category sampled. Moreover, we gain something in terms of speed, indeed, while one should consider all the base categories in the classical MNP and average the results, a single iteration of the sMNP is needed which is p times faster (with equally optimized algorithms). Moreover the sMNP represents a proper Bayesian paradigm which automatically incorporates posterior uncertainty about the base category and it should be preferred.

In our case, it is worth noticing that, even if the behaviour of the b parameter may not be desirable in terms of mixing, which can be also witnessed by the not so high acceptance rates of the $\widetilde{w}_{ij}$, the obtained results should be positively considered since all the indicators seems to show a proper quality from the point of view of computation correctness. In fact, as underlined before, we have ideals *"fat hairy caterpillar"* shaped traceplots and also the autocorrelation graphs can lead to the very same conclusion.

Overall the results of the sMNP seem promising as a tool to overcome the classical Multinomial Probit Model. The improving in accuracy and time of computation suggest that this is the path to follow to retrieve more stable inferences and behaviours, thus it appears clear to us that the profuse effort to implement the new algorithm has been profitable and rewarding.

References

[1] James H. Albert and Siddhartha Chib. Bayesian analysis of binary and polychotomous response data. *Journal of the American Statistical Association*, 88(422):669–679, 1993.

[2] Lane F. Burgette, David Puelz, and P. Richard Hahn. A Symmetric Prior for Multinomial Probit Models. *Bayesian Analysis*, 16(3):991 – 1008, 2021.

[3] Kosuke Imai and David van Dyk. A bayesian analysis of the multinomial probit model using marginal data augmentation. *Journal of Econometrics*, 124:311–334, 02 2005.

[4] Xiyun Jiao and David van Dyk. A corrected and more efficient suite of mcmc sampler for the multinomial probit model. *arXiv preprint arXiv:1504.07823*, 2015.

[5] Robert McCulloch and Peter Rossi. An exact likelihood analysis of the multinomial probit model. *Journal of Econometrics*, 64(1-2):207–240, 1994.

[6] David van Dyk and Kosuke Imai. Mnp: R package for fitting the multinomial probit model. *Journal of Statistical Software*, 14, 05 2005.

Appendix: details of the algorithm

```
Step 0: Initialize $t = 0$, $W^{(0)}$, $\beta^{(0)}$, $\alpha^{(0)}$, $\Sigma^{(0)}$, $b^{(0)}$.
while $t < T$ do:
    Step 1: Remove the elements corresponding to $b^{(t)}$ to obtain $X_b$, $\widetilde{W}_b$, $\widetilde{\beta}_b$, $\widetilde{\Sigma}_b$
    Step 2: Update $\widetilde{W}$
        for $i = 1, ..., n$ do:
            for $j = 1, ..., p$ do:
                Sample $\widetilde{w}_{ji} \sim \mathcal{TN}(\mu, V)$
            end for
        end for
        Derive $\widetilde{W}$ from $\widetilde{W}_b$ inserting $b$-th row
    Step 3: Update $\widetilde{\beta}$
        Compute $\hat{\beta}_b = [\sum_{i=1}^n X_{i,b}^T \Sigma_b^{-1} X_{i,b} + A^{-1}]^{-1} [\sum_{i=1}^n X_{i,b}^T \Sigma_b^{-1} \widetilde{W}_{i,b}]$
        Sample $\widetilde{\beta}_b \sim \mathcal{N}(\hat{\beta}_b, \alpha^2 (\sum_{i=1}^n X_{i,b}^T \Sigma_b^{-1} X_{i,b} + A^{-1})^{-1})$
        Derive $\widetilde{\beta}$ from $\widetilde{\beta}_b$ inserting $b$-th element
    Step 4: Update $b$, $\Sigma_b$ and $\alpha$
        Sample $b$ from a Multinomial distribution s.t.
            $p(b \,|\, \widetilde{\beta}, \widetilde{W}) \propto |S_b + \sum_i (\widetilde{W}_{i,b} - X_{i,b}\widetilde{\beta}_b)(\widetilde{W}_{i,b} - X_{i,b}\widetilde{\beta}_b)^T|^{-(n+\nu_b)/2}$
        Sample $\widetilde{\Sigma}_b \sim$ inv-Wishart$(n + \nu_b, S_b + \sum_i (\widetilde{W}_{i,b} - X_{i,b}\widetilde{\beta}_b)(\widetilde{W}_{i,b} - X_{i,b}\widetilde{\beta}_b)^T)$
        Compute $\alpha = \sqrt{\frac{\mathrm{tr}(\widetilde{\Sigma}_b)}{p-1}}$
        Compute $\Sigma_b = \widetilde{\Sigma}_b / \alpha^2$
    Step 5: Compute $W = \widetilde{W}/\alpha$ and $\beta = \widetilde{\beta}/\alpha$
    $t = t + 1$
end while
```