Serialized Interacting Mixed Membership Stochastic Block Model

Gael POUX-MÉDARD
ERIC Lab
Université de Lyon, 69361
Lyon, France
0000-0002-0103-8778

Julien VELCIN
ERIC Lab
Université de Lyon, 69361
Lyon, France
0000-0002-2262-045X

Sabine LOUDCHER
ERIC Lab
Université de Lyon, 69361
Lyon, France
0000-0002-0494-0169

Abstract—Last years have seen a regain of interest for the use of stochastic block modeling (SBM) in recommender systems. These models are seen as a flexible alternative to tensor decomposition techniques that are able to handle labeled data. Recent works proposed to tackle discrete recommendation problems via SBMs by considering larger contexts as input data and by adding second order interactions between contexts’ related elements. In this work, we show that these models are all special cases of a single global framework: the Serialized Interacting Mixed membership Stochastic Block Model (SIMSBM). It allows to model an arbitrarily large context as well as an arbitrarily high order of interactions. We demonstrate that SIMSBM generalizes several recent SBM-based baselines. Besides, we demonstrate that our formulation allows for an increased predictive power on five real-world datasets.

I. INTRODUCTION

Clustering is a core concept of machine learning. Among other applications, it has proven to be especially fit to tackle real-world recommendation problems. A recommendation consists in guessing an output entity based on a given context. This context can often be represented as a high dimensional set of input entities. On retail websites for instance, the context could be the ID of a user, the last product she bought, the last visited page, the current month, and so on. Clustering algorithms look for regularities in these datasets to reduce the dimensionality of the input context to its most defining characteristics. Continuing the online retail example, a well designed algorithm would spot that a mouse, a keyboard and a computer screen are somehow related buys, and that the next buy is likely to be another computer device. Besides, when given a set of users, subsets of users are likely to have similar interest in a given product if their buying history is similar. We can define such groups of people that share similar behaviors using clustering algorithms; this is called collaborative filtering. One of the most widely used approaches to perform this task relies on tensor decomposition.

Tensor decomposition approaches provide a variety of efficient mathematical tools for breaking a tensor into a combination of smaller components. The most popular tensor decomposition methods are Non-negative Matrix Factorization (NMF) and Tensor rank (or Canonical Polyadic) CP decomposition. However, decomposition methods are based on linear decomposition of a real-valued tensor $D$, which is unfit to tackle discrete problems. These methods can efficiently infer continuous outputs (the rating of a movie as in [1] for instance, or the number of buys of a product) but must be tweaked in order to consider discrete outputs (the next buy on an online retail website for instance). In this case, a possible approach consists in mapping all possible discrete outputs as a continuous variable. This is straightforward as in the case of movie ratings, because the set of possible ratings ($1, 2, 3, ...$) can be ordered on a continuous scale. However, if one wants to recommend one of several products (mouse, keyboard, computer, ...), the mapping of possible outputs to
that represents possible input entities, given a context. 

\[ R = \sum_{k} \theta_{f_n,k} = 1 \forall f_n \] (1)

Once the nodes membership is known, the SIMSBM infers the clusters multipartite network, whose weighted hyper-edges stand for the probability of an output \( o \) given a combination of clusters. The hyper-edge corresponding to the clustered context \( \vec{k} = (k_1, \ldots, k_N) \) associated to the output \( o \) is written 

\[ p_{\vec{k}}(o) \in \mathbb{R}^{K_{a(f_1)} \times \ldots \times K_{a(f_N)} \times O} \]

Note that the clustered context \( \vec{k} \) can take any value among all the possible permutations \( \vec{K} = \{ K_1, \ldots, K_N \} \). As we want SIMSBM to infer a distribution over possible outputs in a given context, the edges of the multipartite graph are related by the following constraint: 

\[ \sum_{o} p_{\vec{k}}(o) = 1 \forall \vec{k} \in \vec{K} \] (2)

Finally, the probability of an output \( o \) given a context of input entities \( \vec{f} \) can be written as: 

\[ P(o|\vec{f}) = \sum_{\vec{k} \in \vec{K}} p_{\vec{k}}(o) \prod_{n \in N} \theta_{f_n,k_n}^{(a(f_n))} \] (3)

From Eq.3, we can define the log-likelihood of the model as: 

\[ \ell = \sum_{(\vec{f}, o) \in \mathcal{R}} \log \left( \sum_{\vec{k} \in \vec{K}} p_{\vec{k}}(o) \prod_{n \in N} \theta_{f_n,k_n}^{(a(f_n))} \right) \] (4)

A. Inference

In this section, we derive an Expectation-Maximization algorithm for inferring the model’s parameters \( p, \theta \). Such algorithm guarantees the convergence towards a local maximum of the likelihood function [6].
TABLE I

| NOTATIONS |
|-----------------|
| $f_n$           | An input entity, can take any value in $F_n$ |
| $F_n$           | Set of possible input entities for layer $n$ |
| $\vec{f}$      | Context vector $(f_1, ..., f_N)$ |
| $o$             | Output entity, can take any value in $O$ |
| $N$             | Number of input layers $|\vec{f}|$ |
| $R^v$           | Data, a list of $(N+1)$-plets $(f_1, ..., f_N, o)$ |
| $a(f_n)$        | Type of entity $f_n$ |
| $K_{a(f_n)}$    | Number of available clusters for type $a(f_n)$ |
| $\theta(a(f_n))$ | Membership matrix for entities of type $a(f_n)$ |
| $p(o)$         | Clusters’ multi-partite network for output $o$ |
| $K$            | All clusters permutations $\{k_1, ..., k_N\}$ |
| $\vec{k}$      | One permutation of clusters $\{k_1, ..., k_N\}$ in $K$ |
| $c_0(x)$       | Count of element $x$ in vector $\vec{v}$ |
| $C_{f_n}$      | Total count of $f_n$ in $R^v$ |

1) E-step: Using Jensen’s inequality, we can rewrite Eq.4 as:

$$\ell = \sum_{(f,o) \in R^v} \log \left( \prod_{k \in K} p_k(o) \prod_{n \in N} \frac{\theta(a(f_n))}{\omega_{f,o}(k)} \right)$$

(5)

The inequality holds as an equality when:

$$\omega_{f,o}(k) = \frac{p_k(o) \prod_{n \in N} \theta(a(f_n))}{\sum_{k' \in K} p_{k'}(o) \prod_{n \in N} \theta(a(f_n))}$$

(6)

Eq.6 constitutes the expectation step of the EM algorithm. This derivation is intended as a fast way of deriving the correct result. An alternative method that is more explicit about the underlying concepts used in the derivation is described in the Supplementary Material; the final expression for $\omega_{f,o}(k)$ is however identical.

2) M-step: We take back Eq.4 and add Lagrangian multipliers $\phi$ to account for the constraints on $\theta$. We maximize of the resulting constrained likelihood $\ell_c$ according to each latent variable:

$$\frac{\partial \ell_c}{\partial \phi_{a(m)}(i)} = \frac{\partial}{\partial \phi_{a(m)}(i)} \left[ \ell - \sum_k \phi_k(i) \left( \sum_k g_{ik} - 1 \right) \right]$$

$$\Leftrightarrow \phi_{a(m)}(i) = \sum_{(f,o) \in \partial m} \omega_{f,o}(k) c_k(n)$$

(7)

The term $c_k(n)$ arises because of the non-linearity induced by the interaction between input entities of the same type. Let $i_m$ be the indices where entity $m$ appears in the input vector $\vec{f}$. The corresponding entries of the permutation vector $k$ are noted $k_m$. Then, $c_k(n) = \{1\}k_m = \sum_{i \in i_m} n$ is the count of $n$ in $k_m$. When $n$ appears $c_k(n)$ times in a permutation comprising $k_m$, so will a term log $\theta_{a(m)}(k)$, whose derivative is $c_k(n)\theta_{a(m)}$, hence this term arising. Note that $c_k(n) = 0$ nullifies the contribution of permutations $\vec{k}$ where $n$ does not appear in $k_m$. Therefore we can restrict the sum over $\vec{k}$ in Eq.7 to the set $\partial_m = \{k|k \in K, n \in k_m\}$. We also defined the set $\partial_m = \{(f,o)|(f,o) \in R^v, o \in \partial_m\}$.

Using Eq.1 and Eq.7, we compute $\phi_{m}(a(m))$:

$$\sum_n \phi_{m}(a(m)) = \sum_{k \in K} c_k(n) \omega_{f,o}(\vec{k})$$

(8)

When summing over $n$, $c_k(n)$ successively counts the number of times each $n$ appears in $k_m$, which equals the length of $k_m$. Therefore $\sum n_i = k_m = c(f,m)$ is the number of times input entity $m$ appears in the entry $(f,o)$ considered, which does not depend on $k$. $c_m$ is the total count of $m$ in the dataset. Note that this differs from the derivation proposed in [3], where nonlinear terms are not accounted for.

The derivation of the maximization equation for $p$ is very similar. Let $\partial_s = \{(f,o)|(f,o) \in R^v, o = s\}$. We solve:

$$\frac{\partial \ell_s}{\partial p_{f,o}(s)} = \frac{\partial}{\partial p_{f,o}(s)} \left[ \ell - \sum_k \psi_k \left( \sum_o p_k(o) - 1 \right) \right] = 0$$

$$\Leftrightarrow \psi_{f,o}(s) = \sum_{(f,o) \in \partial s} \omega_{f,o}(\vec{r}) p_{f,o}(s)$$

(9)

Finally, combining Eq.7 with Eq.8, and the two last lines of Eq.9, the maximization equations are:

$$\begin{cases} 
\phi_{m}(a(m)) = \sum_{(f,o) \in \partial m} \omega_{f,o}(k) c_k(n) \\
\psi_{f,o}(s) = \sum_{(f,o) \in \partial s} \omega_{f,o}(\vec{r}) p_{f,o}(s)
\end{cases}$$

(10)

From Eq.10 we can show that for a given number of clusters for each type $(K_{a(f_1)} ..., K_{a(f_N)})$, one iteration of the EM algorithm runs in $O(|R^v|)$. Note however that $|R^v|$ grows exponentially with the context length.

We must define a nomenclature to refer to each special case of the SIMSBM—what input entity types are considered, and how many interactions for each type. We use the notation SIMSBM(number interactions type 1, number interactions type 2, ...). For instance, SIMSBM(2,3) represents a case where the SIMSBM considers two types of input entities, with the first one interacting as pairs with other entities of same type, and the second one interacting as triples with entities of the same type. The corresponding data has a shape $(f_1, f_2, g_1, g_2, g_3, o)$ where $f$ and $g$ are the two considered types.
III. BACKGROUND ON STOCHASTIC BLOCK MODELS

A. Existing works

As stated in the introduction, recent years saw a growing interest for Stochastic Block Models (SBM) to tackle collaborative filtering problems in recommender systems [2]–[4]. These models first cluster input entities together, and then analyze how these clusters relate to each other. Each input entity can be associated either to one cluster only (single-membership SBM) [7]–[9] or to a distribution over available clusters (Mixed Membership SBM, or MMSBM) [10]. While the single-membership SBM has been successfully applied to a range of problems [8], [9], [11], [12], inference is done using greedy algorithms, typically simulated annealing, making it unfit for large scale real-world applications [2].

The Mixed-Membership SBM (MMSBM) is a major extension of Single-Membership SBM that has been proposed in the seminal work [10]. In the frame of collaborative filtering for recommender systems, [2] proposed a bipartite network extension. This model has later been extended to consider triposes of input entities instead of pairs [3]. It assumes that all the entities in a given triple are linked together by a given relation. This boils down to assuming data can be represented in the form of a tripartite network instead of a bipartite network. Another extension of [10] proposes to consider the case of input entities of the same type, modeled as a bipartite graph [4]. This is relevant when trying to guess an output given a pair of input entities of same type – that is when entities carry the same semantic meaning. When considering input entities of the same type, one must consider symmetric clustering; the probability of an output based on the entities pair (A, B) should not differ from the probability of an output based on (B, A). The authors solve the problem by clustering both entities using a same membership matrix, whose components then interact with each other. This differs from other recent works on interaction modeling that do not consider clustering [13], [14] or the non-linearity induced by symmetric interactions [3], [15].

B. SIMSBM generalizes several state-of-the-art models

The formulation of SIMSBM allows to recover several state-of-the-art works. Each of these previous models was introduced as different and novel in their respective publications, whereas they could be presented as simple iterations of SIMSBM instead. Building on this generalization, SIMSBM provides a degree of modeling flexibility that goes beyond the existing literature – modeling arbitrarily large context sizes and interaction order.

Now, we briefly show how our formulation allows to recover several state-of-the-art models. Throughout this section, we denote input entities of different types by different letters (e.g. \( f_1 \) is not of the same type as \( g_1 \)), and the model’s output as \( o \). The set of corresponding membership matrices for each type is noted as \( \Theta = \{ \theta^{(f)}, \theta^{(g)} \} \) and one edge of the multipartite clusters–interaction tensor is noted \( \{ p_{k(f_1),k(f_2),k(g)}(o) \} \) where \( k(f) \) is one of the possible cluster indices for an input entity of type \( f \).

1) Bi-MMSBM [2]: The Bi-MMSBM has first been proposed in [2], and has since been applied on several occasions [3], [16]. In this modeling, data is made of triplets \( (f_1, g_1, o) \). Each entity is associated a node on a side of a bipartite graph \( f_1’s \) on one side, \( g_1’s \) on the other and edges represent the probability of an output \( o \). We recover the Bi-MMSBM with our model by setting \( \Theta = \{ \theta^{(f)}, \theta^{(g)} \} \) and the bipartite clusters network tensor \( \{ p_{k(f_1),k(g)}(o) \} \). This correspond to SIMSBM(1,1).

2) T-MBM [3]: The T-MBM is a model proposed in [3] that goes a step further than [2] by adding a layer to the bipartite network used to model quadruplet data \( (f_1, f_2, g_1, o) \). This model aims at modeling interactions between entities of same type as in [4] by clustering \( f_1 \) and \( f_2 \) using a same membership matrix, but does not account for nonlinear terms. We recover the T-MBM modeling by setting \( \Theta = \{ \theta^{(f)} \} \) and a bipartite clusters network tensor \( \{ p_{k(f_1),k(f_2),k(g)}(o) \} \). Our formulation allows to go further by adding an arbitrary number of layers to the multipartite networks proposed in [2], [3]. This correspond to SIMSBM(2,1).

3) IMMSBM [4]: The IMMSBM proposed in [4] models interactions between entities of the same type to predict an output. The data takes the form \( (f_1, f_2, o) \). Each input entity is still associated to one node on either side of a bipartite graph, except that here the membership matrix is shared between the two layers. The links between each pair of clusters represent the probability of an output \( o \). We recover the IMMSBM with our model by setting \( \Theta = \{ \theta^{(f)} \} \) and the bipartite clusters network tensor \( \{ p_{k(f_1),k(f_2),k(g)}(o) \} \). Importantly, our formulation allows to consider interactions between \( n \) input entities instead of simply pair interactions. This correspond to SIMSBM(2).

4) Indirect generalizations: We did not detail the generalization of other families of block models because our algorithm does not readily fits these cases. However, it is worth mentioning that MMSBM has been developed as an alternative to Single Membership SBM [17] that allows more flexibility [10]. Our model reduces to most existing SBM by modifying the definition of the entries of \( \theta^{(a)} \). In the Single Membership SBM, Eq.1 reads \( \theta^{(a)}_{f_2,k} = \delta_{k,x} \) where \( x \) is one of the \( K_a \) possible values for \( k \) and \( \delta \) is the Kronecker’s delta. This means the membership vector of each input entity equals 1 for one cluster only, and 0 anywhere else. Therefore, the optimization process is not the same as we described. In practice optimization is done with greedy algorithms such as the simulated annealing [12], [16].

IV. EXPERIMENTS

A. Datasets and evaluation protocol

1) Datasets: The datasets we consider here are summarized in Table II and detailed as Supplementary Material. They are made of tuples representing the context and the associated observed output. Each of them is made available along with the implementation of our model on GitHub2.

2) Datasets and implementation available at https://github.com/GaelPouxMedard/SIMSBM

1146
90% of each dataset’s documents are used as a training set, and the other 10% are used as an evaluation set. Each iteration of the SIMSBM is run 100 times on every dataset. The EM algorithm stops once the relative variation of the likelihood falls below $10^{-4}$ for 30 iterations in a row. We present the average results over all the runs. The number of clusters has been chosen based on the existing literature on similar datasets (Imdb [2], MrBanks [16], Spotify and PubMed [4]).

Finally, when SIMSBM is evaluated on a dataset containing more interactions than it is designed to consider, the model is trained on the lower-order corresponding dataset. For instance, imagine a dataset considering one type interacting three times. This dataset is made of one observation only $(1, 2, 3)$. A SIMSBM iteration that considers pair interactions will then be trained on triplets $(1, 2, o), (1, 3, o)$ and $(2, 3, o)$, and evaluation will be performed accordingly.

2) Baselines and evaluation: Evaluation is done according to the maximum F1 score, the precision at 1 ($P@1$), the area under the ROC (UCROC), the area under the precision-recall curve (UCPR), the F1 score with a 10% threshold (vgF1), the precision at 1 (vgPrec), the recall at 1 (vgRec), the mean average precision (vMap), the average precision at 10 positions (vAP10), the average precision over all positions (vAP), the mean absolute error (vErr), and the mean absolute percentage error (vErrNorm). Evaluation is done according to the maximum F1 score, the precision at 1 ($P@1$), the area under the ROC (UCROC), the area under the precision-recall curve (UCPR), the F1 score with a 10% threshold (vgF1), the precision at 1 (vgPrec), the recall at 1 (vgRec), the mean average precision (vMap), the average precision at 10 positions (vAP10), the average precision over all positions (vAP), the mean absolute error (vErr), and the mean absolute percentage error (vErrNorm).
under the ROC curve, the area under the Precision-Recall curve (or Average Precision); since the problem is about multi-label classification, we consider the weighted version of these metrics –metrics are computed individually for each class, and averaged with each classes’ weight being equal to the number of true instances in the class. The results are averaged over all 100 runs. We also consider the rank average precision and the normalized covering error (only here lower is better). These last two metrics account for label ranking performance.

We compare to several baselines: a naive baseline (predicts outputs according to their frequency), Naive Bayes, K-nearest neighbors, NMF, Tensor Factorization [18], MMSBM [10], Bipartite-MMSBM [2], IMMSBM [4] and T-MBM [3].

B. Discussion

We present our main results in Table III. In this table, we see that our formulation systematically outperforms the proposed baselines, as well as the ones it generalizes. In most cases, the possibility to add a layer or to consider higher-order interactions improves the performance over the existing baselines (MMSBM, Bi-MMSBM, IMMSBM and T-MBM). We further demonstrate the usefulness of SIMSBM by running replication studies for [2] and [16] that are provided as Supplementary Material; we get similar results as these previous works. About the Spotify dataset in Table III, note that artists are often added to a playlist in a row, leading to the probability of the next artist being the same as the one immediately before her to be higher. Adding interaction terms adds noise in the modeling. This explains why the triple interactions version of SIMSBM does not perform better than its pair-interactions [2] or no-interaction [10] iterations.

Besides numerical results, Table III underlines how easy the SIMSBM makes it to design tailored models for a variety of different problems. Using a single framework, we could consider input context sizes ranging from 1 to 4 entries, and interaction orders going from 1 to 3. Model selection (or in this case, SIMSBM iteration selection) becomes simpler as a result, since different input settings can be readily tested. Up to now, adding a new layer (e.g. T-MBM with respect to Bi-MMSBM) or increasing the interaction order (e.g. IMMSBM w.r.t MMSBM) required a new dedicated work to explain the advance; SIMSBM unifies these approaches into one global model, each iteration of which is readily usable in a number of applications, as shown in Table III. To underline this, we ran also ran replication studies of two recent papers [16], [19], given as Supplementary Material.

V. CONCLUSION

In this paper, we developed a global framework, SIMSBM, that generalizes several existing models from the literature. We derived an expectation-maximization algorithm that runs in linear time with the number of observations. We then demonstrated that SIMSBM recovers several models from the literature as special cases, such as Bi-MMSBM, IMMSBM and T-MBM. This results in a highly flexible model that can be applied to a broad range of classification problems, as shown through systematic evaluation of the proposed formulation on several real-world datasets. In particular, we cited throughout the text a number of experimental studies conducted in medicine, social behaviour and recommendation using special cases of our model; using alternative iterations of the SIMSBM framework may help further improve the description and understanding of the interacting processes at stake between an arbitrary greater number of input entities.

REFERENCES

[1] Y. Koren, R. Bell, and C. Volinsky, “Matrix factorization techniques for recommender systems,” Computer, vol. 42, no. 8, pp. 30–37, 2009.
[2] A. Godoy-Lorite, R. Guimerà, C. Moore, and M. Sales-Pardo, “Accurate and scalable social recommendation using mixed-membership stochastic block models,” PNAS, vol. 113, no. 50, pp. 14 207–14 212, 2016.
[3] M. Tarrès-Deulofeu, A. Godoy-Lorite, R. Guimerà, and M. Sales-Pardo, “Tensorial and bipartite block models for link prediction in layered networks and temporal networks,” Phys. Rev. E, vol. 99, p. 032307, Mar 2019.
[4] G. Poux-Médard, J. Velicin, and S. Loudcher, “Interactions in information spread: quantification and interpretation using stochastic block models,” RecSys ’21, 2021. [Online]. Available: doi:10.1145/3460231.3474254
[5] G. Poux-Médard, “Interactions in information spread,” WWW ’22: Companion Proceedings of the Web Conference 2022, pp. 313–317, 2022.
[6] R. M. Neal and G. E. Hinton, A View of the Em Algorithm That Justifies Incremental, Sparse, and other Variants. Dordrecht: Springer Netherlands, 1998, pp. 355–368. [Online]. Available: https://doi.org/10.1007/978-94-011-5014-9_12
[7] P. W. Holland, K. B. Laskey, and S. Leinhardt, “Stochastic blockmodels: First steps,” Social Networks, vol. 5, no. 2, pp. 109–137, 1983. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0378873389900217
[8] R. Guimerà, A. Llorente, and M. Sales-Pardo, “Predicting human preferences using the block structure of complex social networks,” PLOS One, vol. 7, no. 9, 2012.
[9] T. Funke and T. Becker, “Stochastic block models: A comparison of variants and inference methods,” PloS one, 2019.
[10] E. Airolidi, D. Blei, S. Fienberg, and E. Xing, “Mixed membership stochastic blockmodels,” Journal of Machine Learning Research, vol. 9, pp. 1991–1992, 2008.
[11] R. Guimerà and M. Sales-Pardo, “A network inference method for large-scale unsupervised identification of novel drug-drug interactions,” PLoS Comput Biol, 2013.
[12] D. J. Cobo-López S., Godoy-Lorite A., “Optimal prediction of decisions and model selection in social dilemmas using block models,” EPJ Data Sci, vol. 7(48), 2018.
[13] E. Christakopoulou and G. Karypis, “Hoslim: Higher-order sparse linear method for top-n recommender systems,” PAKDD, 2014. [Online]. Available: 10.1007/978-3-642-55014-9_4
[14] H. Steck and D. Liang, “Negative interactions for improved collaborative filtering: Don’t go deeper, go higher,” RecSys ’21, p. 34–43, 2021. [Online]. Available: 10.1145/3460231.3474273
[15] S. A. Myers and J. Leskovec, “Clash of the contagions: Cooperation and competition in information diffusion,” 2012 IEEE 12th International Conference on Data Mining, pp. 539–548, 2012.
[16] G. Poux-Médard, S. Cobo-Lopez, J. Duch, R. Guimerà, and M. Sales-Pardo, “Complex decision-making strategies in a stock market experiment explained as the combination of few simple strategies,” EPJ Data Science, vol. 10, 2021.
[17] Y. J. W. Y. Wong, “Stochastic blockmodels for directed graphs,” Journal of the American Statistical Association, vol. 82, no. 397, pp. 8–19, 1987.
[18] A. Karatzoglou, X. Amatriain, L. Baltrunas, and N. Oliver, “Multiverse recommendation: N-dimensional tensor factorization for context-aware collaborative filtering,” in Proceedings of the Fourth ACM Conference on Recommender Systems, ser. RecSys ’10. New York, NY, USA: Association for Computing Machinery, 2010, p. 79–86. [Online]. Available: https://doi.org/10.1145/1864708.1864727
[19] M. Gutiérres-Roig, C. Segura, J. Dutch, and J. Perello, “Market imitation and win-stay lose-shift strategies emerge as unintended patterns in market direction guesses,” PLOS One, 2016.