ESANN 2023 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 4-6 October 2023, i6doc.com publ., ISBN 978-2-87587-088-9. Available from http://www.i6doc.com/en/.

Deep dynamic co-clustering of streams of count data: a new online Zip-dLBM

Giulia Marchello[1], Marco Corneli[1,2], Charles Bouveyron[1]

1- Université Côte d'Azur, Inria, CNRS, Laboratoire J.A.Dieudonné,
Maasai team, Nice, France.

2- Université Côte d'Azur, Laboratoire CEPAM, Nice, France.

Abstract. Co-clustering is a technique used to analyze complex and high-dimensional data in various fields. However, traditional co-clustering methods are usually limited to dense data sets and require massive amount of memory, which can be limiting in some applications. To address this issue, we propose an online co-clustering model that processes the data incrementally and introduces a novel latent block model for sparse data matrices. The proposed model employs a LSTM neural network and a time and block dependent mixture of zero-inflated distributions to model sparsity and aims to detect real-time changes in dynamics through Bayesian online change point detection. An original variational procedure is proposed for inference. Simulations demonstrate the effectiveness of the methodology for count data.

1 Introduction

As the amount of data in various fields grows exponentially, techniques as clustering to summarize data are also increasingly needed. Co-clustering is useful in this context since it clusters both observations and features simultaneously, providing useful data summaries. Furthermore, there is a growing need to develop machine learning models for time-dependent sparse data matrices and, although many notable methods have been introduced in this field in recent decades (Marchello et al., 2022b; Casa et al., 2021), the development of online co-clustering methods remains largely unexplored. This paper proposes an online extension of Zip-dLBM (Marchello et al., 2022a). We introduce three novelties in this regard. The first is the ability of the estimation algorithm to work online with streams of data. The second is the addition of an online change point detection method. By capturing the data's dynamic behavior, the method can identify abnormal events that affect the generative process. To detect these changes we make use of the Bayesian Online Change Point Detection (BOCPD, Adams and MacKay, 2007) that runs on the estimated model parameters in real time. The third novelty relies on a different modeling choice for the time evolving parameters, in fact fully connected neural networks are substituted with LSTMs, as their structure is deemed more appropriate for the purpose. Therefore, this model introduces a new approach by incorporating an online inference method for Zip-dLBM and online changing point detection.

2 Zip-dLBM

The following section reminds the Zero-Inflated Poisson Dynamic Latent Block Model (Zip-dLBM, Marchello et al., 2022a) for batch processing. This paper has been proposed in a general way for any Zero-Inflated distribution, however, here we focus on the Zero-Inflated Poisson (ZIP) distribution. In Zip-dLBM the observed data are assumed to be collected into time evolving matrices, over the interval $[0, T]$. Being in a discrete time setting, we assume a time partition of equally spaced points: $0 = t_0 < t_1 < t_u \leq t_U = T$. With a slight abuse of notation we denote by t the time point t_u. At time t, the incidence matrix $X(t) \in \mathbb{N}^{N \times M}$ has $X_{ij}(t)$ as generic element that counts the number of interactions between the observation i and feature j that took place between t and $t-1$.

Clusters modeling The rows and columns of $X(t)$ are clustered into Q and L groups, respectively. We denote by $Z(t) := \{Z_{iq}(t)\}_{i \in 1,\ldots,N; q \in 1,\ldots,Q}$ the latent matrix representing the clustering of N rows into Q groups at a given time point t. We assume that the i-th row of $Z(t)$ follows an evolving multinomial distribution, parameterized by $\alpha(t) := (\alpha_1(t), \ldots, \alpha_Q(t))$. In a similar fashion, we introduce a latent matrix $W(t) \in \{0,1\}^{M \times L}$, labeling the column clusters at time t, and whose j-th row $W_j(t)$ follows a multinomial distribution of parameter $\beta(t) := (\beta_1(t), \ldots, \beta_L(t))$. The two random matrices Z and W are further assumed to be independent.

Sparsity Modeling In order to model a potentially extreme sparsity, the observed data are assumed to follow a mixture of block-conditional Zero-Inflated Poisson (ZIP) distributions, where the entries $X_{ij}(t)$ are conditionally independent: $X_{ij}(t)|Z_i(t), W_j(t) \sim ZIP(\Lambda_{Z_i(t),W_j(t)}, \pi(t))$. We denote as Λ the $Q \times L$ block-dependent intensity function of the Poisson distribution $\mathcal{P}(X_{ij}(t), \cdot)$, and $\pi(t)$ is a vector of length T that indicates the level of sparsity at any given time period. We finally provide an equivalent formulation of the above equation in terms of a hidden random matrix, $A \in \{0,1\}^{N \times M}$, where, independently for all i and j, we define $A_{ij}(t) \sim \mathcal{B}(\pi(t))$. Here, $\mathcal{B}(\cdot)$ denotes the Bernoulli distribution of parameter $\pi(t)$. Thus:

$$\begin{aligned} &A_{ij}(t) = 1 \Rightarrow X_{ij}(t)|Z_i(t), W_j(t) = 0 \\ &A_{ij}(t) = 0 \Rightarrow X_{ij}(t)|Z_i(t), W_j(t) \sim \mathcal{P}(X_{ij}(t), \Lambda_{Z_i(t),W_j(t)}). \end{aligned} \tag{1}$$

Modeling the temporal evolution of the parameters We assume that the mixing proportions $\alpha(t)$, $\beta(t)$, and the sparsity parameter $\pi(t)$ are governed by a system of ordinary differential equations (ODEs). Since we work in discrete time we discretize the dynamic systems by making use of their Euler scheme:

$$\begin{cases} a(t+1) = a(t) + f_Z(a(t)), \\ b(t+1) = b(t) + f_W(b(t)), \\ c(t+1) = c(t) + f_A(c(t)). \end{cases} \tag{2}$$

ESANN 2023 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 4-6 October 2023, i6doc.com publ., ISBN 978-2-87587-088-9. Available from http://www.i6doc.com/en/.

where f_Z, f_W and f_A are three continuously differentiable functions and the parameters $\alpha(t)$, $\beta(t)$ and $\pi(t)$ are softmax transformations of $a(t)$, $b(t)$ and $c(t)$, respectively.

2.1 The joint distribution

The set of the model parameters is denoted by $\theta = (\Lambda, \alpha(t), \beta(t), \pi(t))$ and the latent variables used so far are $Z(t)$, $W(t)$, and $A(t)$. Thus, the likelihood of the complete data reads:

$$p(X, Z, W, A|\theta) = p(X|Z, W, A, \Lambda, \pi)p(A \mid \pi)p(Z|\alpha)p(W|\beta). \tag{3}$$

The terms on the right hand side of the above equation can be further developed, for details see Marchello et al. (2022a).

3 Online inference for stream data

In this section, we present Stream Zip-dLBM. The objective is to perform co-clustering of rows and columns in real-time as new data becomes available. To prevent memory overload, we have revisited the inference algorithm of Zip-dLBM, enabling data to be processed without the need to store it in memory. To allow the algorithm to update parameter estimations continuously as new data is incorporated, we use a moving window, $G_d(t)$, of size d. Using a moving window allows us to keep only part of the data in memory. Therefore, for example, in t, we keep in memory only the data in the interval $[t-d, t]$ that will be used for parameters estimation, while the data before the interval can be discarded from the model. This allows to prevent memory overloads and maintain the algorithm's functionality.

3.1 Inference

Since we cannot compute the joint conditional distribution, $p(A, Z, W|X, \theta)$, we rely on a variational procedure which optimizes a lower bound of the log-likelihood. Let us thus introduce a variational distribution $q(.)$ in order to decompose the log-likelihood as follows:

$$\log p(X|\theta) = \mathcal{L}(q; \theta) + KL(q(.)||p(.|X, \theta)), \tag{4}$$

where $\mathcal{L}$ denotes a lower bound and KL the Kullaback-Liebler divergence between the true and the approximate posterior. The objective is to find a distribution $q(.)$ that maximizes the lower bound $\mathcal{L}(q, \theta)$. In order to allow the optimization of $\mathcal{L}(q, \theta)$, we further assume that $q(A(t), Z(t), W(t))$ factorizes as follows:

$$q(A(t), Z(t), W(t)) = q(A(t))q(Z(t))q(W(t)) = \prod_{i=1}^{N}\prod_{j=1}^{M} q(A_{ij}(t)) \prod_{i=1}^{N} q(Z_i(t)) \prod_{j=1}^{M} q(W_j(t)). \tag{5}$$

3.2 VE-Step

The optimal variational updates of $q(\cdot)$, under the assumption in Eq. (5), can be obtained as in (Bishop, 2006, Ch. 10). We denote by $\delta_{ij}(t) := q(A_{ij}(t) = 1)$ the variational probability of success for $A_{ij}(t)$, $\tau_{iq}(t) := q(Z_{iq}(t) = 1)$ the variational probability of success of $Z_{iq}(t)$, and $\eta_{j\ell}(t) := q(W_{j\ell}(t) = 1)$ the variational probability of success of $W_{j\ell}(t)$. The explicit updating equations and the proofs are provided in Marchello et al. (2022a).

3.3 Variational M-Step

Although the lower bound, $\mathcal{L}(q;\theta)$, can be explicitly computed, it is here omitted for lack of space. From that bound, we can optimize the model parameters θ, while keeping $q(\cdot)$ fixed.

3.3.1 Update of Λ

Here our goal is to derive the online update of the Zero-inflated Poisson intensity parameter, Λ. The variational distribution $q(A, Z, W)$ is kept fixed, while the lower bound is maximized with respect to Λ at every time instant t, to obtain its update, $\hat{\Lambda}$. Hence, we compute the derivative of the lower bound $\mathcal{L}(q, \theta)$ with respect to Λ obtaining:

$$\begin{aligned}\hat{\Lambda}_{q\ell} &= \frac{\sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{u=1}^{(t-1)} \tau_{iq}(u)\eta_{j\ell}(u)\Big(X_{ij}(u) - \delta_{ij}(u)X_{ij}(u)\Big) + \sum_{i=1}^{N}\sum_{j=1}^{M} \tau_{iq}(t)\eta_{j\ell}(t)\Big(X_{ij}(t) - \delta_{ij}(t)X_{ij}(t)\Big)}{\sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{u=1}^{(t-1)} \tau_{iq}(u)\eta_{j\ell}(u)\Big(1 - \delta_{ij}(u)\Big) + \sum_{i=1}^{N}\sum_{j=1}^{M} \tau_{iq}(t)\eta_{j\ell}(t)\Big(1 - \delta_{ij}(t)\Big)} \\ &= \frac{N_{q\ell}^{old} + N_{q\ell}^{(t)}}{D_{q\ell}^{old} + D_{q\ell}^{(t)}} = \frac{N_{q\ell}^{old}}{D_{q\ell}^{old} + D_{q\ell}^{(t)}} + \frac{N_{q\ell}^{(t)}}{D_{q\ell}^{old} + D_{q\ell}^{(t)}}.\end{aligned}$$

Now we can distinguish between a part known at time step $t-1$, namely $N_{q\ell}^{old}$ and $D_{q\ell}^{old}$, and the current updates at time t, $N_{q\ell}^{(t)}$ and $D_{q\ell}^{(t)}$. Then, denoting $\hat{\Lambda}_{q\ell}^{old} = N_{q\ell}^{old}/D_{q\ell}^{old}$, we obtain the final online update:

$$\hat{\Lambda}_{q\ell} = \hat{\Lambda}_{q\ell}^{old} \cdot \frac{D_{q\ell}^{old}}{D_{q\ell}^{old} + D_{q\ell}^{(t)}} + \frac{N_{q\ell}^{(t)}}{D_{q\ell}^{old} + D_{q\ell}^{(t)}}. \tag{6}$$

3.3.2 Update of α*,* β *and* π *through deep neural networks*

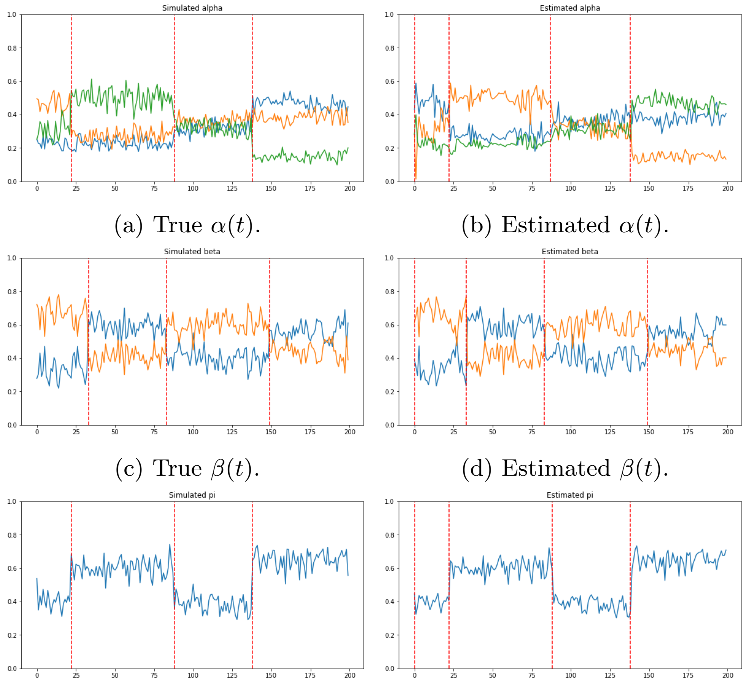
As mentioned in Section 2, the mixture proportions α and β, as well as the sparsity parameter π are driven by three systems of differential equations. As we assumed that the functions f_A, f_W and f_Z are continuous, we propose to parametrize them with three LSTM networks (Hochreiter and Schmidhuber, 1997). LSTM operates on a sequence of a specific length, and it produces a sequence of the same length, but shifted one time step ahead. In Stream Zip-dLBM the sequence length has the same size of the moving window, $G_d(t)$. For instance, at current time t, the input of LSTM would consist of a series of values ranging from $t-1-d$ to $t-1$, while the output will be a sequence of predicted values from $t-d$ to t. Therefore, in the initial stage of the algorithm (i.e. the first d time points), the parameters $\alpha(t)$, $\beta(t)$, and $\pi(t)$ are modeled via two-layer fully connected neural networks. At $t = d + 1$, the estimates from the previous time step, obtained via fully connected networks, serve as input to LSTM, which is used for online parameter estimation from this point on. Once the neural nets are trained via back-propagation (SGD) they provide us with the current ML estimates of $\alpha(t)$, $\beta(t)$ and $\pi(t)$.

4 Bayesian online change point detection

As previously stated, one of the aims of Stream Zip-dLBM is to perform online change point detection. To accomplish this task, we combine the Bayesian Online Change Point Detection (BOCD) method, proposed in a seminal paper by Adams and MacKay (2007), with our strategy. BOCD detects change points based on the estimation of the posterior distribution over the current "run length", or time since the last change point, given the data so far observed, using a simple message-passing algorithm. Essentially, the run length is used to determine if a new data point belongs to the current partition based on previous observations. If the new data point belongs to the current partition, the run length will increase by 1 at the next time step, otherwise it will reset to 0. This process is continuously repeated at each time step. It is worth noticing that the BOCD algorithm is typically implemented in an online fashion, analyzing the data as it streams in. However, in our case, we directly apply the algorithm to the estimates of $\alpha(t)$, $\beta(t)$, and $\pi(t)$ that are generated by the LSTM. To prevent detecting change points on parameters that will be recalculated in future time steps, we run the BOCD algorithm only on time points "behind" $G_d(t)$. Stated differently, at time t, BOCD runs on parameter values at time instants $t - d$.

5 Experiments on simulated data

The purpose of this section is to highlight the main features of the proposed model over a simulated data set. We aim at demonstrating the validity of the inference algorithm presented in the previous sections. A simulated data set with dimension $350 \times 300 \times 200$ has been generated to perform this experiment. The simulated dynamics of $\alpha(t)$, $\beta(t)$ and $\pi(t)$ can be seen on the left-hand side of Figure 1. Based on the mixture proportions, the values of the latent variables were then simulated through their distributions. Next, we used the sparsity proportions, $\pi(t)$, and the intensity function, Λ, to simulate the three-dimensional tensor X as Zero-Inflated Poisson variables. We then applied the Stream Zip-dLBM model to the simulated data set, using the actual values of $Q = 3$ and $L = 2$ to demonstrate the model's ability to recover the parameters. Figure 1 displays the true mixture proportions on the left side and the online estimates on the right side. The red dashed lines depict the simulated and estimated change points, respectively. We can see that Stream Zip-dLBM perfectly recovers the evolution of the model parameters over time, including the change points. These results suggest promising applications of this model in real-world data analysis.



(a) True $\alpha(t)$.

(b) Estimated $\alpha(t)$.

(c) True $\beta(t)$.

(d) Estimated $\beta(t)$.

(e) True $\pi(t)$.

(f) Estimated $\pi(t)$.

Fig. 1: Evolution of the true (left) and estimated (right) proportions of the parameters $\alpha(t)$, $\beta(t)$ and $\pi(t)$, respectively.

References

Adams, R. P. and MacKay, D. J. (2007). Bayesian online changepoint detection. *arXiv preprint arXiv:0710.3742*.

Bishop, C. M. (2006). Approximate inference. pages 461–517. Springer-Verlag, Berlin, Heidelberg.

Casa, A., Bouveyron, C., Erosheva, E., and Menardi, G. (2021). Co-clustering of time-dependent data via the shape invariant model. *Journal of Classification*, 38(3):626–649.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Marchello, G., Corneli, M., and Bouveyron, C. (2022a). A deep dynamic latent block model for the co-clustering of zero-inflated data matrices.

Marchello, G., Fresse, A., Corneli, M., and Bouveyron, C. (2022b). Co-clustering of evolving count matrices with the dynamic latent block model: application to pharmacovigilance. *Statistics and Computing*, 32(3):1–22.