Supplementary material for the paper: Co-clustering through Optimal Transport

Charlotte Laclau [1] **Ievgen Redko** [2] **Basarab Matei** [1] **Younès Bennani** [1] **Vincent Brault** [3]

In this supplementary material, we present a couple of complementary elements which were omitted in the main paper. We first introduce the Sinkhorn-Knopp algorithm, then we explain how exactly the synthetic data sets were simulated. Finally, we analyse the running time results obtained for both `CCOT` and `CCOT-GW` on the generated data sets.

1. Sinkhorn-Knopp algorithm

For the sake of completeness, we first present the Sinkhorn's theorem and explain how it was used to derive the solution of the regularized optimal transport.

Theorem ((Sinkhorn & Knopp, 1967))**.** *If A is an $n \times n$ matrix with strictly positive elements, then there exist diagonal matrices D_1 and D_2 with strictly positive diagonal elements such that $D_1 * A * D_2$ is doubly stochastic. The matrices D_1 and D_2 are unique modulo multiplying the first matrix by a positive number and dividing the second one by the same number.*

We can now cite the following result.

Lemma ((Cuturi, 2013), Lemma 2)**.** *For $\lambda > 0$, the solution γ^*_λ is unique and has the form*

$$\gamma^*_\lambda = diag(\boldsymbol{\alpha})\xi_\lambda diag(\boldsymbol{\beta}),$$

where $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are two non-negative vectors of $\mathbb{R}^d$ uniquely defined up to a multiplicative factor and $\xi_\lambda = e^{-\lambda M}$ is the element-wise exponential of $-\lambda M$.

According to (Cuturi, 2013), the form of the solution presented in this Lemma has already been known in the optimal transportation theory (Erlander & Stewart, 1990). Now since $\xi_\lambda = e^{-\lambda M}$ is strictly positive, Sinkhorn's theorem suggests that there exists a unique (up to rescaling) doubly-stochastic matrix that has the desired form $\text{diag}(\boldsymbol{\alpha})\xi_\lambda\text{diag}(\boldsymbol{\beta})$. Finally, this matrix can be found using the iterative procedure known as Sinkhorn-Knopp algorithm defined as follows:

$$\boldsymbol{\alpha} \leftarrow 1./\xi_\lambda\boldsymbol{\beta}, \qquad \boldsymbol{\beta} \leftarrow 1./\xi'_\lambda\boldsymbol{\alpha}.$$

2. Additional experimental results

In this Section, we describe the generative process used to obtain the synthetic data sets. After that, we analyse the impact of the hyper-parameters on our methods and the running their running time results.

2.1. Simulation process

As mentioned in the paper, we simulate data following the generative process of the Gaussian Latent Block Models. These models rely on the assumption that for each block, the elements of the data matrix $\mathcal{A} = a_{ij}, i = 1, \ldots, n; j = 1, \ldots, d$ are distributed according to a Gaussian distribution $\mathcal{N}(\mu_{k\ell}, \sigma^2_{k\ell})$ with $\mu_{k\ell} \in \mathbb{R}$ and $\sigma^2_{k\ell} \in \mathbb{R}^+$, $k = 1, \ldots, g; \ell = 1, \ldots, m$, following a probability density function of this form

$$f(\mathcal{A};\Theta) = \sum_{(\mathbf{z},\mathbf{w})\in\mathcal{Z}\times\mathcal{W}} \prod_{i,k} \pi_k^{z_{ik}} \prod_{j,\ell} \rho_\ell^{w_{j\ell}} \prod_{i,j,k,\ell} p(a_{ij};(\mu_{k\ell},\sigma^2_{h\ell}))$$

where

$$p(a_{ij};(\mu_{k\ell},\sigma^2_{k\ell})) = \frac{1}{\sqrt{2\pi\sigma^2_{k\ell}}} \exp - \left\{\frac{(x_{ij}-\mu_{k\ell})^2}{2\sigma^2_{k\ell}}\right\}^{z_{ik}w_{j\ell}}.$$

Therefore, this model is parametrised by $\Theta = (\pi, \rho, \delta)$ where $\pi = (\pi_1, \ldots, \pi_m)$, $\rho = (\rho_1, \ldots, \rho_m)$ and $\delta = ((\mu_{11}, \sigma^2_{11}), \ldots, (\mu_{gm}, \sigma^2_{gm}))$.

Then, the simulation process is as follows

- **Input**: n, d, g, m, Θ.
  1. Simulate $\mathbf{z}$ according to a multinomial distribution with parameters $(1, \pi_1, \ldots, \pi_g)$.
  2. Simulate $\mathbf{w}$ according to a multinomial distribution with parameters $(1, \rho_1, \ldots, \rho_m)$
  3. Simulate each co-cluster $\mathcal{A}_{k\ell}$ according to Gaussian density with $(\mu_{k\ell}, \sigma^2_{k\ell})$

[1] CNRS, LIPN, Université Paris 13 - Sorbonne Paris Cité, France [2] CNRS UMR 5220 - INSERM U1206, Univ. Lyon 1, INSA Lyon, F-69621 Villeurbanne, France [3] CNRS, LJK, Univ. Grenoble-Alpes, France. Correspondence to: Charlotte Laclau <charlotte.laclauc@univ-grenoble-alpes.fr>.

Proceedings of the 34^{th} International Conference on Machine Learning, Sydney, Australia, PMLR 70, 2017. Copyright 2017 by the author(s).

[0] The first author of this paper is now a post-doc in CNRS, LIG, Univ. Grenoble-Alpes, France

- **Output** Data matrix $\mathcal{A}$, partitions $\mathbf{z}$ and $\mathbf{w}$.

For the sake of reproducibility, we also report the parameters used in order to generate D1, D2, D3 and D4 in Table 1.

2.2. Visualisation of α and β for **CCOT-GW**

As the main paper only presents the visualization of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ for `CCOT`, we present the same result for the kernelized version, `CCOT-GW` for D4 in Figure 1.

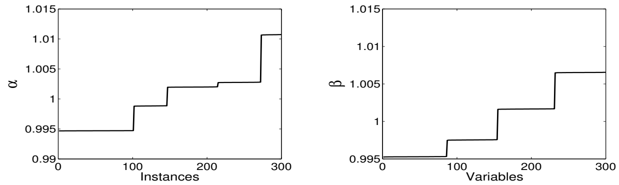

Figure 1. Visualisation of α and β obtained with `CCOT-GW` on D4 which have a 5×4 blocks structure and unbalanced clusters for instances.

These vectors correctly reveal 4 and 3 significant jumps corresponding to 5 and 4 clusters, respectively.

2.3. Impact of n_s and λ

The proposed algorithms require as input the number of desired subsamples, n_s (for `CCOT` only) and the value of the regularization parameter, λ. From Figure 2, one can see that for all four data sets the co-clustering error stabilizes when n_s reaches approximately 700.

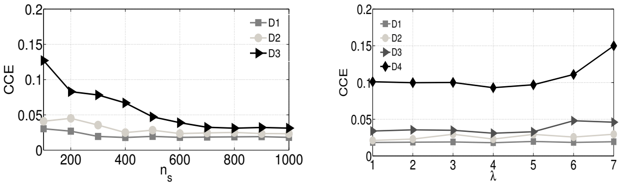

Figure 2. On the left, the CCE as a function of the number of random samples n_s and on the right as a function of the regularization parameter λ.

Since D3 has the most complicated block structure, the obtained results are also more sensible to the value of n_s compared to other data sets and require a greater number of samplings. Regarding λ, we observe very slight variations of the CCE for D1 and D2. However, for D3 higher values of this parameter impact negatively the performance of our method. This can be explained by the fact that increasing λ accentuates the differences between the values of $\boldsymbol{\alpha}$ (resp. $\boldsymbol{\beta}$). By doing so, small gaps, that correspond to overlapping clusters, tend to merge leading to less accurate results. Regarding the regularization parameter, the same observation is valid for `CCOT-GW`.

2.4. Running time complexity

The running time performance of our algorithms was evaluated on a cluster machine Intel(R) Xeon(R) CPU X2637 @ 3.00GHz. We report the average running time (in seconds) of both approaches for 100 trials obtained on the generated data sets in Figure 3.

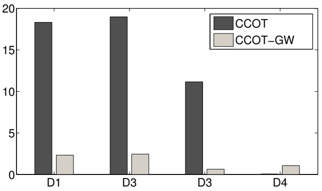

Figure 3. Mean running times expressed in seconds, over 100 trials, of `CCOT` and `CCOT-GW` for each synthetic data sets.

We observe that even though the theoretical complexity of `CCOT` is more interesting, it suffers from the data sampling that is required to cluster all data instances and features. This limitation becomes less and less pronounced as the number of data instances approaches the number of features. Indeed, on D4, where no sampling is required, `CCOT` only takes approximately 0.06 seconds to accurately produce the resulting partitions, which is significantly faster than `CCOT-GW`. For all other data configurations, the kernelized `CCOT-GW` algorithm is faster than `CCOT`.

2.5. Recommendation on MOVIELENS

The main task on MOVIELENS is to recommend movies to users that might fit their interests. For a given co-clustering structure, this can be done by recommending movies to users based on the ratings provided by users who belong to the same cluster. In order to evaluate the efficiency of our approach for this task, we propose to use 90% of the available ratings for training purpose, and the remaining 10% for testing. Since our goal is to predict if a user likes a given movie or not without specifying the degree of the preference, we assume that a rating above 3 stands for movies that were liked. In order to estimate the ratings in the testing phase, we calculate the mean of the block obtained during the training phase and attribute it to the missing values picked for the testing. After 10-folds cross-validation, we obtain that for 89% of the testing values, our approach is able to correctly identify the taste of the users. This shows its potential for recommendation systems application.

Table 1. Value of the parameters used for the simulations.

Data	Proportions	σ	μ	Data	Proportions	σ	μ
D1	$\pi = \rho = \begin{pmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{pmatrix}$	$\sigma = 0.1, \quad \forall k, \ell$	$\mu = \begin{pmatrix} 4.0 & 0.5 & 1.5 \\ 1.8 & 4.5 & 1.1 \\ 1.5 & 1.5 & 5.5 \end{pmatrix}$	D3	$\pi = \begin{pmatrix} 0.5 & 0.5 \end{pmatrix}$ $\rho = \begin{pmatrix} 0.5 & 0.2 & 0.1 & 0.2 \end{pmatrix}$	$\sigma = 0.2 \quad \forall k, \ell$	$\mu = \begin{pmatrix} 4.0 & 0.5 & 7.5 & 0.5 \\ 0.5 & 3.5 & 7.8 & 0.5 \end{pmatrix}$
D2	$\pi = \rho = \begin{pmatrix} 0.2 & 0.3 & 0.5 \end{pmatrix}$	$\sigma = 0.15 \quad \forall k, \ell$	$\mu = \begin{pmatrix} 4.0 & 0.5 & 1.5 \\ 1.8 & 4.5 & 5.1 \\ 3.5 & 1.5 & 5.5 \end{pmatrix}$	D4	$\pi = \begin{pmatrix} 0.1 & 0.2 & 0.2 & 0.3 & 0.2 \end{pmatrix}$ $\rho = \begin{pmatrix} 0.25 & 0.25 & 0.25 & 0.25 \end{pmatrix}$	$\sigma = 0.15 \quad \forall k, \ell$	$\mu = \begin{pmatrix} 1.5 & 1.5 & 1.5 & 1.5 \\ 2.5 & 1.5 & 1.5 & 1.5 \\ 2.6 & 2.6 & 1.5 & 1.5 \\ 2.4 & 2.5 & 2.6 & 2.5 \end{pmatrix}$

References

Cuturi, Marco. Sinkhorn distances: Lightspeed computation of optimal transport. In *Proceedings NIPS*, pp. 2292–2300, 2013.

Erlander, Sven and Stewart, Neil F. *The Gravity model in transportation analysis : theory and extensions*. Topics in transportation. VSP, Utrecht, The Netherlands, 1990.

Sinkhorn, Richard and Knopp, Paul. Concerning nonnegative matrices and doubly stochastic matrices. *Pacific Journal of Mathematics*, 21:343–348, 1967.