Supplementary Material for Robust Bayesian Tensor Factorization with Zero-Inflated Poisson Model and Consensus Aggregation

Anonymous Author(s)
Affiliation
Address
email

Contents

S1 Zero-inflated Poisson tensor factorization (ZIPTF)

S1.1 Zero-inflated Poisson model

Let $\mathcal{X}$ be a count data in $\mathbb{R}^{I_1 \times I_2 \ldots \times I_N}$. We define the index set $\overline{I}$ as the collection of all possible indices, i.e., $\overline{I} = \{i_1 i_2 \ldots i_N \ : \ 1 \leq i_j \leq I_j, \ 1 \leq j \leq N\}$. We say $\mathcal{X}$ has Zero-inflated Poisson (ZIP) distribution if for every $I \in \overline{I}$:

$$P(\mathcal{X}_I = x_I) = p_I \mathbb{1}_{x_I = 0} + (1 - p_I) \frac{e^{-\lambda} \lambda^{x_I}}{x_I!}, \tag{1}$$

where the outcome variable x_I has non-negative integer values, λ_I is the expected Poisson count, and p_I is the *probability of extra zeros* [11]. As an abbreviation, we write it as $\mathcal{X}_I \sim ZIP(\lambda_I, p_I)$. The ZIP can be considered as the *product* of a Poisson random variable $\mathcal{Y}_I \sim Poisson(\lambda_I)$ and an independent Bernoulli variable $\Phi_I \sim Bernoulli(p_I)$ [4]. The Bernoulli variable Φ_I takes the value of 1 when $\mathcal{X}_I$ is equal to 0, due to the Bernoulli component, and takes the value of 0 otherwise.

We consider the low rank $R \geq 1$ decomposition of the zero-inflated count tensor $\mathcal{X}$:

$$\mathcal{X} \approx \sum_{r=1}^{R} a_r^{(1)} \otimes a_r^{(2)} \otimes \cdots \otimes a_r^{(N)}. \tag{2}$$

Submitted to 37th Conference on Neural Information Processing Systems (NeurIPS 2023). Do not distribute.

Hence, for $I = i_1 i_2 \dots i_N$, the reconstruction $\sum_{r=1}^{R} a_{i_1 r}^{(1)} a_{i_2 r}^{(2)} \dots a_{i N r}^{(N)}$ can be interpreted as the mean of the distribution from which the observed count $\mathcal{X}_I$ is assumed to be sampled. Then we have:

$$\mathcal{X}_I \sim ZIP(\lambda_I = \sum_{r=1}^{R} a_{i_1 r}^{(1)} a_{i_2 r}^{(2)} \dots a_{i N r}^{(N)}, p_I). \tag{3}$$

S1.2 Variational Inference for ZIPTF

For given position $I = i_1 i_2 \dots i_N$, we consider the rank R decomposition in Eqn. (3). In Bayesian Poisson factorizations, the Gamma distribution is utilized as a prior to induce sparsity, and it is assumed that each latent factor matrix $A^{(k)} = [a_1^{(k)} \dots a_R^{(k)}] \in \mathbb{R}_+^{I_k \times R}$, $1 \le k \le N$, follows a Gamma distribution [3, 14]. Therefore, for each $a_{jr}^{(k)}$ in Eqn. (3), we can approximate it as follows:

$$a_{jr}^{(k)} \sim Gamma(\alpha^{(k)}, \beta^{(k)}), \quad 1 \le k \le N, \tag{4}$$

where $\alpha^{(k)} > 0$ and $\beta^{(k)} > 0$ represent the shape and rate parameters of the distribution, with the expectation $E[a_{jr}^{(k)}] = \frac{\alpha^{(k)}}{\beta^{(k)}}$ and $Var[a_{jr}^{(k)}] = \frac{\alpha^{(k)}}{\beta^{(k)^2}}$. Additionally, for ZIP models a latent variable ξ is introduced to capture the hidden state of the probability of extra zeros which specify $\Phi \sim Bernoulli(p_I)$ [16, 2]. Let $S(.)$ denote the *logistic sigmoid* function, given by $S(x) = \frac{1}{1+e^{-x}}$, then:

$$\xi = S(\zeta) \text{ where } \zeta \sim Normal(\mu, \sigma). \tag{5}$$

Let $Z = \{A^{(1)}, A^{(2)}, \dots, A^{(N)}, \Phi\}$, consider the posterior distribution $P(Z|\mathcal{X}, \mathcal{H})$, given a model hyperparameter set $\mathcal{H} = \{\alpha^{(1)}, \beta^{(1)}, \alpha^{(2)}, \dots, \beta^{(2)}, \dots, \alpha^{(N)}, \beta^{(N)}, \mu, \sigma\}$.

Variational inference approximates the true posterior distribution using a family of probability distributions $\mathcal{Q}$ over hidden variables [2]. This family of distributions is characterized by free parameters, and the key assumption is that each latent variable is independently distributed given these parameters. We assume a variational family of distributions $\mathcal{Q}$ indexed by a set of variational parameters $V = \{\gamma^{(1)}, \delta^{(1)}, \gamma^{(2)}, \delta^{(2)}, \dots, \gamma^{(N)}, \delta^{(N)}, \overline{\mu}, \overline{\sigma}\}$ where $(\gamma^{(k)}, \delta^{(k)})$ are variational shape and rate parameters of the Gamma distribution for the latent factor along the $k-$th mode, and $(\overline{\mu}, \overline{\sigma})$ are the variational parameters for ζ. We use a fully factorized mean-field approximation [2] and the variational distribution factors as the following:

$$\mathcal{Q}(A^{(1)}, A^{(2)}, \dots, A^{(N)}, \Phi) = \mathcal{Q}(\Phi; \overline{\mu}, \overline{\sigma}) \prod_{k=1}^{N} \mathcal{Q}(A^{(k)}; \gamma^{(k)}, \delta^{(k)}). \tag{6}$$

where $a_{jr}^{(k)} \sim Gamma(\gamma_{jr}^{(k)}, \delta_{jr}^{(k)})$ and $\Phi_I \sim Bernoulli(S(\zeta))$ for $\zeta \sim Normal(\overline{\mu}, \overline{\sigma})$. The goal is to choose a member q^* of the variational family variational distributions which minimizes the Kullback-Leibler (KL) divergence of the exact posterior from $\mathcal{Q}$:

$$q^*(Z) = \arg \min_{q(Z) \in \mathcal{Q}} D_{KL}\left(q(Z) \| P(Z|\mathcal{X}, \mathcal{H})\right). \tag{7}$$

Upon examining the KL divergence, we encounter a significant challenge: it involves the true posterior distribution $P(Z|\mathcal{X}, \mathcal{H})$, which is not known. Nevertheless, we can rewrite the KL divergence as follows:

$$\begin{aligned}
D_{KL}\left(q(Z) \| P(Z|\mathcal{X}, \mathcal{H})\right) &= \int q(Z) \log\left(\frac{q(Z)}{P(Z|\mathcal{X}, \mathcal{H})}\right) dZ \quad ‘ & (8)\\
&= \int q(Z) \log\left(\frac{q(Z) P(\mathcal{X}, \mathcal{H})}{P(Z, \mathcal{X}, \mathcal{H})}\right) dZ & (9)\\
&= \log\left(P(\mathcal{X}, \mathcal{H})\right) \int q(Z) dZ - \int q(Z) \log\left(\frac{P(Z, \mathcal{X}, \mathcal{H})}{q(Z)}\right) dZ & (10)\\
&= \log\left(P(\mathcal{X}, \mathcal{H})\right) - \int q(Z) \log\left(\frac{P(Z, \mathcal{X}, \mathcal{H})}{q(Z)}\right) dZ. & (11)
\end{aligned}$$

The second term in Eqn. (11) is called Evidence Lower Bound (ELBO). We know that the KL divergence is non-negative, therefore, $\log\left(P(\mathcal{X},\mathcal{H})\right) \geq \text{ ELBO}(q(Z)) = \int q(Z)\log\left(\frac{P(Z,\mathcal{X},\mathcal{H}}{q(Z)}\right) dZ$.

$$\begin{aligned} \text{ELBO}(q(Z)) &= \int q(Z)\log\left(P(Z,\mathcal{X},\mathcal{H})\right) dZ - \int q(Z)\log\left(q(Z)\right) dZ \quad (12) \\ &= E_{q(Z)}[\log\left(P(\mathcal{X},Z,\mathcal{H})\right] - E_{q(Z)}[\log q(Z)]. \quad (13) \end{aligned}$$

The evidence lower bound serves as a transformative tool that converts intractable inference problems into optimization problems that can be tackled using gradient-based methods [2].

Coordinate ascent algorithms are frequently employed in maximizing the evidence lower bound (ELBO)[2, 16]. However, these algorithms require tedious gradient calculations and may not scale well for very large datasets [6, 12]. Closed-form coordinate-ascent updates are applicable to conditionally conjugate exponential family models, but they necessitate analytic computation of various expectations for each new model[6, 12].

Stochastic Variational Inference (SVI) [6] offers a more efficient algorithm by incorporating stochastic optimization [13]. This technique involves utilizing noisy estimates of the gradient of the objective function. To maximize the evidence lower bound (ELBO), we employ a stochastic optimization algorithm known as the *Black Box Inference Algorithm* [12]. This algorithm operates by stochastically optimizing the variational objective using Monte Carlo samples from the variational distribution to compute the noisy gradient (see Section 2, [12] for details). By doing so, it effectively alleviates the burden of analytic computations and provides a more efficient approach to ELBO maximization.

S2 Implementation

S2.1 Implementation of ZIPTF and C-ZIPTF

We present a Python implementation of a versatile Bayesian Tensor Factorization method using Variational Inference. Our implementation leverages Pyro [1], a probabilistic programming framework built on PyTorch. The code for our implementation is available in the supplemental material as well as in the public GitHub repository [link omitted for double-blind review]. The BayesianCP class inherits from `torch.nn.Module` and offers functionalities for model fitting and summarizing the posterior distribution of factor matrices. During model fitting, Stochastic Variational Inference (SVI) is employed with an Adam optimizer [8, 6]. The current implementation supports three models: Zero Inflated Poisson model (ZIPTF), a Gamma Poisson model (GPTF) [14], and a Truncated Gaussian model (TGTF) [5].

S2.2 Implementation of baseline methods

As mentioned in Section 2.1, we utilize the same implementation for the other Bayesian tensor factorization approaches (Gamma Poisson Bayesian Tensor Factorization and Truncated Gaussian Bayesian Tensor Factorization) as the ZIPTF method and the code is provided in the supplemental material. For the remaining baselines used in our comparisons we use the following implementations:

- **Non-negative Matrix Factorization (NMF)**: We use the Python implementation provided in the scikit-learn package. `https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.non_negative_factorization.html`.
- **Consensus Non-negative Matrix Factorization (cNMF)**: We use the Python implementation described in [10], and provided on GitHub `https://github.com/dylkot/cNMF/tree/master`
- **Non-negative CP via Alternating-Least Squares (NNCP-ALS)**: We use the Python implementation provided in the Tensorly package. `http://tensorly.org/stable/modules/generated/tensorly.decomposition.non_negative_parafac_hals.html`

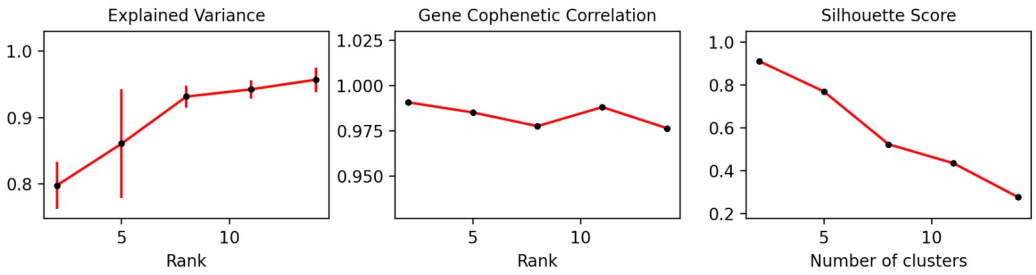

Figure S1: Metrics used in selecting optimal rank for running C-ZIPTF on real single-cell RNA sequencing dataset [7] of immune cells stimulated with interferon beta (IFN-β).

S3 Simulation details

We use a Python adaptation of the Splatter [17] statistical framework given in [10] to simulate single-cell RNA-Seq data. The core of the simulation is a Gamma-Poisson distribution used to generate a cell-by-gene count matrix. While the original Splatter framework supports the simulation of both expression outlier genes and technical dropout (random knockout of counts), the Python adaptation in [10] only keeps outlier expression simulation. Since our method is specifically adapted to handle dropout noise in single-cell data, we add back the modeling of dropout to the Python adaptation. Specifically, after sampling counts from a Poisson distribution, we simulate dropout noise by calculating the probability of a zero for each gene from its mean expression and using that to randomly replace some of the simulated counts with zeros employing a Bernoulli distribution as described in [17].

The distribution of expression values prior to incorporating differential expression was determined based on parameters estimated from a random sample of 8000 cells from an organoid dataset as described in [10]. Specifically, the library size of a cell is sampled from a Lognormal distribution derived from a Normal distribution with a mean of 7.64 and a standard deviation of 0.78. The mean expression of a gene is sampled from a Gamma distribution with a mean of 7.68 and a shape of 0.34. With the probability of 0.00286, a gene will be an outlier from this Gamma distribution and will instead be sampled from a Lognormal distribution derived from a Normal distribution with a mean of 6.15 and standard deviation of 0.49. Additionally, we set a 5% doublet rate. Doublets are formed by randomly sampling a pair of cells, combining their gene counts, and downsampling such that the total count equals the larger of the two.

S4 Real single-cell RNA-Seq data analysis

We obtain the single-cell RNA-Seq data from GEO using accession number GSE96583 `https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE96583`. As described in [7] the dataset contains 14,619 control and 14,446 stimulated cells. As part of the preprocessing step, we filter out multiplets and cells without a cell type assignment. Additionally, we remove samples and cell types that constitute less than 2 percent of cells. After these filtering steps, the dataset contained 14 samples, 7 control and 7 stimulated, and 6 cell types: CD4 T-cells, CD14+ Monocytes, B-cells, CD8 T-cells, NK- cells, FCGR3A+ Monocytes. In order to facilitate biological interpretability of factors and reduce noise in the tensor formed we removed genes that are either not provided with HGNC symbols [15], or had a total count of less than 50 across all cells. Finally, we create a pseudobulk tensor by summing up the raw counts for each cell type, sample, and gene. The resulting pseudobulk data tensor has dimensions S $\times$ C $\times$ G (14 $\times$ 6 $\times$ 9,276), where S, C and G denote the number of samples, cell types and genes respectively. We normalize the tensor such that each sample-cell type pair has a total of 10^6 counts. We first determined the optimal rank for the data by running C-ZIPTF with a range of ranks from 2 to 14 and just 5 restarts. Some of the metrics we considered in deciding the optimal rank including explained variance and silhouette score are shown in Figure S1. We select rank 8 based on these criteria and the full set of factors are shown in Figure S2.

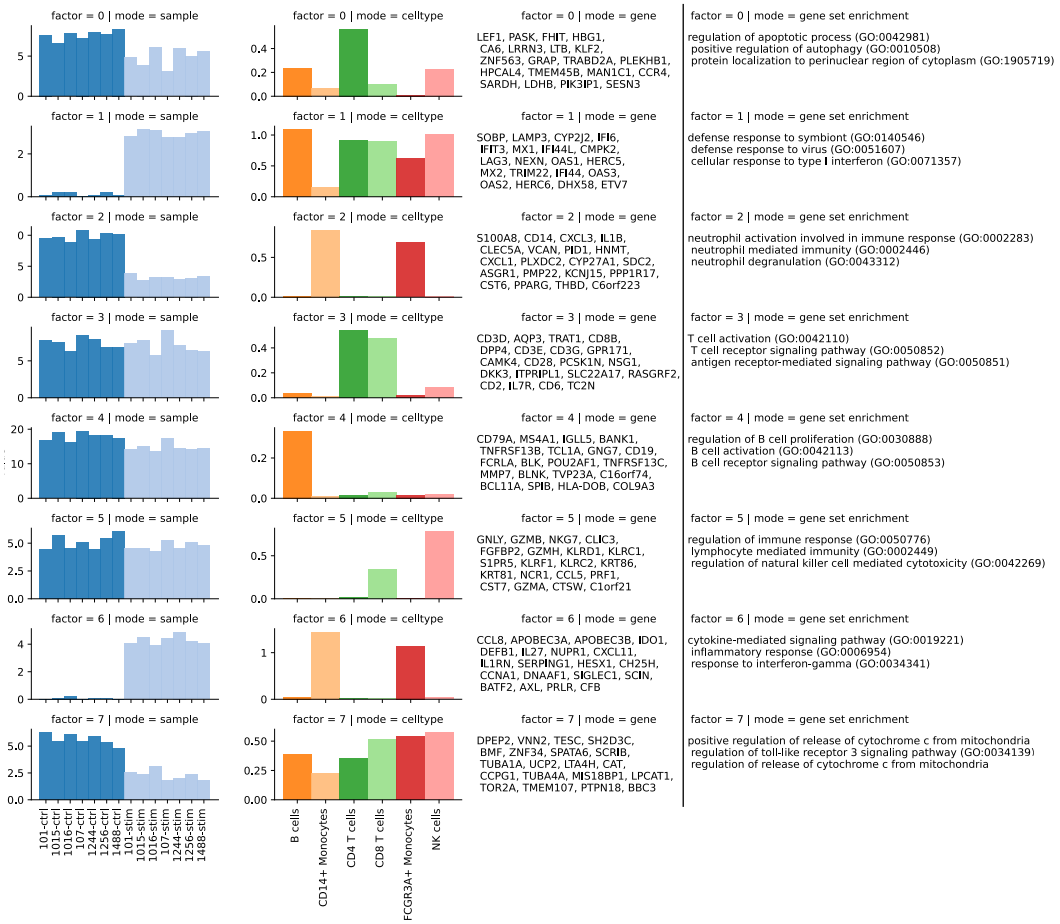

Figure S2: Full set of factors recovered by running C-ZIPTF on real single-cell RNA sequencing dataset [7] of immune cells stimulated with interferon beta (IFN-β). Each row represents a factor, and the first three columns display the three modes: sample, cell type, and gene. The $y-$axis in the sample and cell type modes represent the loading of the sample or cell type on that factor. The gene mode exhibits the top 20 genes associated with the factor. The last column provides the top 3 enriched terms obtained from a gene set enrichment analysis.

References

[1] E. Bingham, J.P. Chen, M. Jankowiak, F. Obermeyer, N. Pradhan, T. Karaletsos, R. Singh, P. Szerlip, P. Horsfall, and N.D. Goodman, N.D, *Pyro: Deep Universal Probabilistic Programming*, Journal of Machine Learning Research (2018), 19(108), 1-6.

[2] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, *Variational Inference: A Review for Statisticians*, Journal of the American Statistical Association (2017), 112:518, 859-877, DOI: 10.1080/01621459.2017.1285773

[3] A. Cemgil, *Bayesian inference for nonnegative matrix factorisation models*, Computational Intelligence and Neuroscience (2009).

[4] M. Chigona and C. Gaetan, *Semiparametric zero-inflated Poisson models with applications to animal abundance studies*, Environmetrics (2007), 18(3), 303-314.

[5] J. L. Hinrich, K. H. Madsen, and M. Mørup, *The probabilistic tensor decomposition toolbox*, Machine Learning: Science and Technology (2020), available online at `https://github.com/JesperLH/prob-tensor-toolbox`.

[6] M.D. Hoffman, D.M. Blei, C. Wang, and J. Paisley,*Stochastic variational inference*(2013), Journal of Machine Learning Research.

[7] H.M Kang, M. Subramaniam, and others, *Multiplexed droplet single-cell RNA-sequencing using natural genetic variation*, Nature biotechnology (2018), 36(1), 89–94, https://doi.org/10.1038/nbt.4042.

[8] D.P. Kingma and J. Ba, *Adam: A method for stochastic optimization* (2014), arXiv preprint arXiv:1412.6980.

[9] T. G. Kolda and B. W. Bader, *Tensor decompositions and applications*, SIAM review **51** (2009), no. 3, 455–500.

[10] D. Kotliar, A. Veres, M. A. Nagy, S. Tabrizi, E. Hodis, D.A. Melton, and P.C. Sabeti, *Identifying gene expression programs of cell-type identity and cellular activity with single-cell RNA-Seq*, eLife (2019), 8, e43803, https://doi.org/10.7554/eLife.43803

[11] D. Lambert, *Zero-Inflated Poisson Regression, with an Application to Defects in Manufacturing*. Technometrics (1992), 34(1):1-14.

[12] R. Ranganath, S. Gerrish, and D. Blei, *Black box variational inference* (2014), Artificial intelligence and statistics, PMLR (pp. 814-822)

[13] H. Robbins and S. Monro,*A stochastic approximation method*(1951), The annals of mathematical statistics, 400-407.

[14] A. Schein, J. Paisley, D. Blei, David, and H. Wallach, *Bayesian Poisson Tensor Factorization for Inferring Multilateral Relations from Sparse Dyadic Event Counts*, Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2015), https://doi.org/10.1145/2783258.278341410.1145/2783258.2783414.

[15] R.L. Seal, B. Braschi, and others, *Genenames.org: the HGNC resources in 2023*, Nucleic Acids Res. PMID: 36243972, DOI:10.1093/nar/gkac888.

[16] M. Simchowitz, *Zero-inflated Poisson factorization for recommendation systems*, Junior Independent Work (advised by D. Blei), Princeton University, Department of Mathematics (2013).

[17] L. Zappia, B. Phipson, and A. Oshlack, *Splatter: simulation of single-cell RNA sequencing data*, Genome Biol (2017),18 174, https://doi.org/10.1186/s13059-017-1305-0.