Simulation Study for CYTOF Model II

Arthur Lui
UC Santa Cruz

December 30, 2017

Contents

1 Introduction and Inferential Goal

I have the following suggestion for the Introduction section. I think you already have all needed components in your sections 1 & 2, but reorganizing the contents and having one Introduction section would make the paper better. Please consider reorganizing as follows;

- Include some scientific backgrounds on the study of NK cell. What are NK cells? Why do researchers study NK cell populations?
- Describe our inferential goals.
- How it was studied before (previous technology)? How the new cytometry technology is better. This may pose some statistical challenges. Discuss them.
- Existing statistical methods for cytometry data. What is missing in the existing statistical methods? Limitations of the existing methods?
- We then introduce our approach in words. How ours is different from the existing? How ours achieves our inferential goals? How ours may be better than the existing ones?
- Whenever it is possible, please include references.

Advances in cytometry has led to more research and greater understanding of natural killer (NK) cells and how their diversity impacts immunity against the development of tumors and other viral diseases. What are NK cells? What are they important?

The main inferential goal of this project is to identify the NK cell phenotypes (or cell-types) in various samples as a set of subpopulations of the set of some provided surface markers. The NK cell-types are latent, and for J markers 2^J different cell-types can be considered.

How do we define NK cell phenotypes? Why do we study NK cell phenotypes? How do we get 2^J different cell types (I think this depends on our definition of cell types)? Please explain how you define a cell type. This may be related to why we use IBP to model cell types as you described the below.

This provides a computational challenge when the number of markers is even moderately large. Thirty-two markers are included in this analysis, and naively enumerating all possible markers is not feasible. We therefore, use a latent feature allocation model to learn the latent structure of predominant cell-types. Latent feature models have been successfully applied to various problems and will be reviewed in section 2.2.

2 Literature Review

2.1 Cytometry

Data for this project is rendered through CyTOF analyses of NK-cell-targeting markers. Having some understanding of CyTOF and NK cells their importance is therefore necessary.

Flow cytometry (developed by Wallace Coulter in the 1950's) is a laser-based biophysical technology for biomarker detection, among other things.In the end, what does it measure? In other words, what do we have in data? We have real values. What do they mean?

It is regularly used to diagnose health disorders like cancer. Cytometry has advanced over the years. Fluorescence-based flow cytometry, which makes use of fluorescent dyes and lasers that emit light at specific wavelengths, is one such advancement that has been mainstream for several decades. In recent years, a new technique called Cytometry at time-of-flight (CyTOF) has surfaced. It makes use of time-of-flight mass spectrometry, where sophisticated devices are used to accelerate, separate, and identify ions by mass. This new method warrants the analysis of multiple parameters in shorter time. Through CyTOF, scientists have been able to better understand natural killer (NK) cells [**?**]. NK cells play critical roles in defending against tumors. Furthermore, their diversity and function are known to be linked. Researchers have thus studied NK cell diversity from various perspectives. For instance, it is known that NK cell diversity is lower at birth [**?**] than in adults. Some researchers have studied the effect of introducing diverse NK cells into tumor patients. Yet again, some researchers have found that patients with higher NK diversity are associated with higher exposure risk of HIV-1, suggesting that existing diversity may decrease flexibility of the antiviral response. Many questions about NK cells remain to be answered. Understanding NK diversity through spectrometry has therefore been an important research area in the bio-sciences. very nice paragraph! but we need to explain the existing statistical methods for cytof data and why the existing methods are not enough for our purpose.

We may shorten the following subsections on IBP and move them to the next section "Probability Model".

2.2 Feature Allocation Models

We desire to learn the latent structure of predominant cell-types, where each cell-type is composed of various combinations of some known markers. Specifically, each cell-type can be represented by a binary vector where a “1” in at a location j of the binary vector indicates that marker j is expressed in the cell-type, and a “0” at the same location indicates that the marker is not expressed. For J distinct markers, 2^J possible cell-types can be constructed. One could create a $J \times 2^J$ matrix which contains all possible cell-types generated by the J markers, with each column containing each cell-type. But this becomes infeasible when J is large. Using Bayesian modelling methods, we can explore the sample space of possible cell-types and learn the predominant ones. For computational efficiency, we require a flexible prior which will also learn the number of cell-types (K) which generate the observed data. The Indian buffet process (IBP) proposed by **?** serves as a suitable prior for modelling the matrix of cell-types. The IBP have been used in a variety of applications where modelling latent features is of interest. These models are also called latent feature allocation models. We will review some of the common representations of the IBP and their computational advantages and properties. We will first discuss the original representation by **?**. Then we will review the stick-breaking construction for the IBP developed by **?**. We will then review the dependent IBP (dIBP) developed by **?** for feature allocation models where prior information on the correlation between the objects (rows) is available.

May include what IBP was used for (such as in specific applications it was used and how it was better). I think this may be too much details for introduction.

2.2.1 The Indian Buffet Process

The Indian buffet process (IBP) proposed by **?** can be constructed by first considering the finite feature allocation model, and then taking the limit with respect to the number of features.

Use your notation instead of using their notation (eg, use J instead of n). Later you will introduce your own notation and readers get confused due to different notation.

For the remainder of this section, let Z be an $n \times K$ binary matrix. The prior probability that object i possessing feature k is π_k. Let π_k have prior distribution $\text{Beta}(\alpha/K, 1)$. For the time being, assume that α is known and fixed.

$$\begin{aligned} \pi_k \mid \alpha &\sim \text{Beta}(\alpha/K, 1) \\ Z_{ik} \mid \pi_k &\sim \text{Bernoulli}(\pi_k) \end{aligned} \tag{1}$$

The matrix Z has an IBP distribution with mass parameter α when the π_k are integrated out and in the limit $K \to \infty$. That is, marginally, $Z \sim \text{IBP}(\alpha)$. While the binary matrix Z is unbounded in the columns, it can be shown that the number non-zero of columns K^+ is distributed $\text{Poisson}(\alpha \sum_{i=1}^{n} \frac{1}{i})$. Moreover, the each row in Z has is expected to have α active features. In other words, the mass parameter α influences the final number of columns in the sampled matrices. This can be shown using law of total expectation:

$$\text{E}\left[\sum_{k=1}^{K} Z_{ik}\right] = \sum_{k=1}^{K} \text{E}\left[Z_{ik}\right] = \sum_{k=1}^{K} \text{E}\left[\text{E}\left[Z_{ik} \mid \pi_k\right]\right] = \sum_{k=1}^{K} \text{E}\left[\pi_k\right] = \sum_{k=1}^{K} \alpha/K = \alpha.$$

Since sampled matrix from model in (1) is extremely sparse, arranging the columns in a way such that columns with more active features are at the left most columns can yield computational advantages. **?** suggest modeling, instead, the equivalence class left-ordered of the binary matrices drawn from the model in (1). That is, columns that represent a higher binary number are at the left of the matrix. The probability mass function (pmf) of the left-ordered matrices can be shown to have the following form:

$$P(\boldsymbol{Z}) = \frac{\alpha^{K_+}}{\prod_{i=1}^{N} K_1{}^{(i)}!} \exp\left\{-\alpha H_N\right\} \prod_{k=1}^{K_+} \frac{(N - m_k)!(m_k - 1)!}{N!}, \tag{2}$$

where $H_N = \sum_{i=1}^{N} i^{-1}$ is the harmonic number, K_+ is the number of non-zero columns in $\boldsymbol{Z}$, m_k is the k^{th} column sum of $\boldsymbol{Z}$, and $K_1^{(i)}$ is the number of features activated in row i of Z that are not activated in previous rows.

The name of this process, like the Chinese restaurant process, suggests a culinary metaphor. The metaphor is as follows. Let Z be an $N \times \infty$ binary matrix. Each row in Z represents a customer who enters an Indian buffet restaurant and each column represents one dish (out of an infinite number of dishes) in the buffet. Customers enter the restaurant one after another. The first customer samples an $r = \text{Poisson}(\alpha)$ number of dishes, where $\alpha > 0$. This is indicated by setting the first r columns of the first row in Z to be 1. The other values in the row are set to 0. Each subsequent customer samples each previously sampled dish with probability proportional to its popularity. That is, the next customer samples dish k with probability m_k/i, where $m_k = \sum_{j=1}^{i-1} Z_{jk}$ is the number of customers that have sampled dish k, and i is the current customer number (or row number in Z). Each customer also samples an additional $\text{Poisson}(\alpha/i)$ number of new dishes. Once all the N customers have gone through this process, the resulting Z matrix will be a draw from the Indian buffet process with mass parameter α.

Other representations and extensions of the IBP have been proposed. I will discuss a few that are relevant to this project.

2.2.2 Stick-breaking Construction for the IBP

The stick-breaking construction for the IBP was proposed by **?**, and can be sampled from by the following scheme:

$$\begin{aligned} v_k \mid \alpha &\sim \text{Beta}(\alpha, 1) \\ \pi_k &:= \prod_{l=1}^{k} v_k \\ Z_{ik} \mid \pi_k &\sim \text{Bernoulli}(\pi_k). \end{aligned} \tag{3}$$

This "stick-breaking" construction is can be derived by first starting with model (1), then ordering the π_k so that $Z_{ik} \mid \pi_{(k)} \sim \text{Bernoulli}(\pi_{(k)})$, where $\pi_{(a)} > \pi_{(b)}$ for all $a < b$, and $a, b \in \mathbb{N}$.

The features in this model are ordered in the sense that activated features tend to appear in the left-most columns. This representation resembles the stick-breaking representation of the Dirichlet process (DP), and so can be extended in similar ways that the DP has been extended. In a Gibbs sampler, the elements in Z can be easily updated using Gibbs steps and metropolis steps.

The number of columns in Z can be fixed in advanced at some large value. But as noted by **?**, the truncation value is somewhat arbitrary. A slice-sampler can be used to avoid this unnecessary approximation. Alternatively, a prior can be placed on K^+, which is what we do in this project.

2.2.3 Dependent IBP

The dependent IBP (dIBP) is one extension of the IBP under the stick-breaking representation which allows prior information on the correlation between items (rows) to be included in feature allocation models. The model is as follow:

cumulative standard Normal distribution function, and $\boldsymbol{S}$ is known and contains the covariance of the objects $i = 1, ..., N$. Note that when $\boldsymbol{S} = \mathbf{I}$ then the dIBP reduces to the stick-breaking construction of the IBP.

why is this dependent? - You have an answer already: prior information on the correlation between items (rows) to be included in feature allocation models through S.

Is the representation with $S = I$ equivalent to the traditional representation of IBP?

2.2.4 Prior for α in Stick-breaking Construction for IBP

In the previous sections, α is treated as fixed and known. A prior distribution can be placed on α to reflect uncertainty. In general, for variants of the IBP that make use of the stick-breaking representation, a (conjugate) Gamma prior can be placed on α, and its full conditional is as follows:

$$\begin{aligned}
v_k|\alpha &\sim \text{Beta}(\alpha, 1) \\
\alpha &\sim \text{Gamma}(a, b), \quad \text{with mean } (a/b) \\
\alpha \mid \boldsymbol{v} &\sim \text{Gamma}(a + K, b - \sum_{k=1}^{K} \log v_k).
\end{aligned}$$

3 Probability Model

I have the following suggestion for the "Probability Model" section. Again I think you already have all needed components but reorganizing the contents would make the paper better. Please consider reorganizing as follows;

- Subsection 1 "Sampling model": Introduce your notation, describe what is observed in data, develop the sampling model. You already have those.
- Subsection 2 "Prior":Describe our prior model
- Subsection 3 "Posterior Simulation": Describe your posterior simulation and how we summarize the posterior samples for inference.

- Whenever it is possible, please include references.

The model and prior specifications are presented in this section. Note that the dIBP will be used in the prior specifications. Some notation which will be used throughout this paper is first presented.

3.1 Notation

Let I represent the number of samples. Let N_i represent the number of cells in sample i, where $i = 1, 2, ..., I$. Let J represent the number of markers. Hence, the raw data $\tilde{y}_{inj}$ represent the raw data for sample i, cell n, and marker j. From a computation perspective, a suitable data structure for the data could be a list (of length I) of matrices of (variable) dimensions $(N_i \times J)$. Let c_{ij} denote the "cutoff" values (provided by the cytometry measuring devices) for sample i, marker j. What does the cutoff do? In other words, why do we need the cutoff? Please explain, e.g., If a value is greater than the cuffoff, the marker is likely to be expressed. Why are they different by i and j?

I would put Y first and then explain we get some missing values and why we get missing values (use Muharram's explanation - such as (i,j) with a missing value implies that marker j in cell i is not expressed with high probability). We then introduce m (missing indicator).

We use such information to model the missing probability given y. Why do we model the missing values? Is it better than discarding? Is it missing at random? or some informative missing? Discuss this. Any reference for modeling missing values?

And define the missingness indicator

$$m_{inj} = \begin{cases} 0, & \text{if } \log\left(\frac{\tilde{y}_{inj}}{c_{ij}}\right) < -\infty \\ 1, & \text{otherwise.} \end{cases}$$

That is, $m_{inj} = 1$ indicates that the expression level **is missing** for sample i, cell n, marker j. Furthermore, define a transformation of the data

$$y_{inj} = \begin{cases} \log\left(\frac{\tilde{y}_{inj}}{c_{ij}}\right), & \text{if } m_{inj} = 0 \\ \text{To be imputed}, & \text{if } m_{inj} = 1. \end{cases}$$

We have all ys regardless of m and only observe y when $m = 0$. That is, some of them are missing (not observed) and the corresponding y is latent. Thus your description here is not quite correct although your equation for the likelihood $p(y, m) = p(y)p(m \mid y)$ is correct.

"To be imputed" is a good explanation between us, but may not be so good for others. We may say "In the Bayesian framework, we treat missing values of y as random variables and impute them..... ".

This transformation will be used in the final model. Note that under this transformation (1) the data have infinite support. (2) $y_{inj} = 0$ has a special meaning, which is that the data take on the same value as the cutoff. Consequently, $y_{inj} > 0$ means that the data take on values greater than the cutoff, etc. (3) y_{inj} for which $\tilde{y}_{inj} = 0$ are regarded as missing, and is to be imputed.

3.2 Model

Based on the notation presented above, we are now ready to present the sampling distribution.

$$\begin{aligned} m_{inj} \mid p_{inj}, y_{inj} &\sim \text{Bernoulli}(p_{inj}) \\ \text{logit}(p_{inj}) &:= \beta_{0ij} - \beta_{1j}\ y_{inj} \\[1em] y_{inj} \mid \mu_{inj}, \gamma_{inj}, \sigma^2_{ij}, \boldsymbol{Z}, \lambda_{in} &\sim \text{Normal}(\mu_{inj}, (\gamma_{inj}+1)\sigma^2_{ij}) \\ \mu_{inj} &:= \mu^*_{Z_{j\lambda_{in}}ij} \\ \gamma_{inj} &:= \gamma^*_{Z_{j\lambda_{in}}ij} \end{aligned} \tag{4}$$

$\text{logit}(p_{inj}) = \beta_{0ij} - \beta_{1j}\ y_{inj}$ What does this mean? Explain.

Your Z has not been formally introduced yet. Your λ_{in} is not introduced yet. We augment the mixture model (which you haven't explained) by introducing the latent cell type indicator.

We discussed cell types earlier. How are cell types related to the parameters in the sampling distribution?

What does γ_{inj} do?

We may move the following to the subsection for posterior computation.

Let $\boldsymbol{\theta}$ represent all parameters (discussed in the next section). Let $\boldsymbol{y}$ represent $y_{inj}\ \forall(i,n,j)$. Let $\boldsymbol{m}$ represent $m_{inj}\ \forall(i,n,j)$.

The resulting **likelihood** is as follows:

$$\begin{aligned} \mathcal{L} = p(\boldsymbol{y}, \boldsymbol{m} \mid \boldsymbol{\theta}) &= p(\boldsymbol{m} \mid \boldsymbol{y}) p(\boldsymbol{y} \mid \boldsymbol{\theta}) \\ &= \prod_{i,n,j} p(m_{inj} \mid y_{inj}, \boldsymbol{\theta}) p(y_{inj} \mid \boldsymbol{\theta}) \end{aligned}$$

$$= \prod_{i,n,j} \left\{ p_{inj}^{m_{inj}} (1-p_{inj})^{1-m_{inj}} \times \frac{1}{\sqrt{2\pi(\gamma_{inj}+1)\sigma^2_{ij}}} \exp\left\{ -\frac{(y_{inj}-\mu_{inj})^2}{2(\gamma_{inj}+1)\sigma^2_{ij}} \right\} \right\}$$

The model is fully specified after priors are placed on all unknown parameters.

3.3 Priors

The specific prior distributions (including hyper-parameters) are included here.

Please explain each of the priors e.g. what does each parameter mean? e.g. what β_{1j} means? Why do we choose the priors? e.g.Why Normal_ for ψ_0? Why do we have μ^*_{0ij}, not μ^*_{kij}? Include texts and references when they are needed.

Instead of putting all in one gigantic equation, we may explain one by one.

Please use letters for fixed hyperparameters. We can explain how we calibrate the priors (how to specify the fixed hyperparameters in the Simulation section).

3.4 Posterior Computation

Discuss the posterior computations. You already have some in your section 4.2. Please move them here and elaborate more.

We will make the number of cell types K random. Discuss how we run MCMC with random K.

Include how to summarize the posterior MCMC samples (which you also explained later). How do we find the posterior estimates of Z, w and other parameters.

Standard MCMC techniques like Gibbs sampling and the Metropolis method are used to sample from the posterior distribution of the parameters. In each of the simulations, 200 samples were gathered after a burn-in of 1200 iterations. The MCMC was thinned by a factor of 5. (i.e. only one of every 5 samples are kept.)

Is this still true? I think that we discussed increasing the number of MCMC iterations to have more samples. Didn't we? Are you going to make a longer run?

4 Simulation Study

We may want to have just sections for simulation studies without subsections, that is, Section 4.1 for Simulation study 1 and Section 4.2 for Simulation study 2 in Section 4 Simulation Studies.

For each section, we include 1) data generation (that you already have in your section 4.1. But, make them specific for each simulation study), 2) fitting the model for the simulated data such as fixed hyperparameters and specifics about MCMC, 3) results inference

As I write later, we include how we specified the fixed hyperparameters and explain why. We also explain how we specify the starting values of the random parameters in "2) fitting the model for the simulated data."

In order to understand the strengths and limitations of the proposed model, two simulation studies were conducted. The studies generate data that resemble the CB CYTOF data. we haven't explained our specific datasets like CB CYTOF data. We don't know the details of CB data. You may write some short paragraph. We will ask Katy to write some paragraphs about the datasets later.

Parameters of greatest interest including Z, μ^*, and W are studied, and the studies are done for different but fixed values for the dimensions of the feature allocation model.

4.1 Data Generation

Data were generated to closely match the proposed model. The steps to simulate data is as follows:

Please explain this in a plain text rather than in "enumerate". Include specific true values used to simulate data.

1. Fix I, J, K.
2. Fix $(\psi_0, \psi_1, \tau_0^2, \tau_1^2)$
    - where $\psi_0 \in (-\infty, 0)$ and $\psi_1, \tau_0^2, \tau_1^2 \in (0, \infty)$
    - For example, the parameters could be (-2, 1, 1, .1). Typically, if $-\psi_0 < \psi_1$ and $\tau_0^2 < \tau_1^2$, the simulated data **will not** resemble real data. So, we should choose $-\psi_0 > \psi_1$ and $\tau_0^2 > \tau_1^2$.
3. Draw σ_{ij}^2 from an inverse gamma distribution. Smaller values of σ_{ij} will yield datasets that are easy to learn from.
4. Draw γ_{0ij}^* from an inverse gamma distribution. This inflates the variance of the observations for which $Z_{j,\lambda_{in}} = 0$.
5. For simplicity, set β_{0ij} to be some negative real value and β_{1j} to be some positive real value. Note that by choosing, each of these values, we implicitly determine the probability that an observation will be treated as missing. Intuitively, β_0 determines the boundary for where observations transition from not-missing to missing. β_1 determines how narrow the boundary is.
6. Fix Z to be some $J \times K$ binary matrix.
    - Ensure that Z does not contain columns that consist of only 0's.

7. Fix W to be an $I \times K$ probability matrix such that each row sums to 1.
    - W_{ik} is the probability that observation y_{inj} takes on cell type k, which corresponds to column k of Z.
8. Set $\lambda_{in} = k$ with probability W_{ik}.
9. Draw μ^*_{0ij} from a Truncated-Normal$(\psi_0, \tau_0^2, -\infty, 0)$.
10. Draw μ^*_{1ij} from a Truncated-Normal$(\psi_1, \tau_1^2, 0, \infty)$.
11. Set $\mu_{inj} = \mu^*_{Z_{j,\lambda_{in}}ij}$.
12. Set $\gamma_{inj} = \gamma^*_{0ij}$ if $Z_{j,\lambda_{in}} = 0$, and 0 otherwise.
13. Draw $\tilde{y}_{inj} \sim \text{Normal}(\mu_{inj}, (1 + \gamma_{inj})\sigma^2_{ij})$.
14. Define $p_{inj} = (1 + \exp\{-\beta_{0ij} + \beta_{1j}\tilde{y}_{inj}\})^{-1}$.
15. With probability p_{inj}, set y_{inj} to be missing, and $\tilde{y}_{inj}$ otherwise.

4.2 MCMC Settings

Please move this section after the prior section (so it becomes Section 3.3 Posterior Computation). Include more details, e.g. discuss some computational challenges and some remedies for those. Include how to summarize the posterior MCMC samples which you explained later. Move the relevant text to this new subsection (Section 3.3 Posterior Computation.

Standard MCMC techniques like Gibbs sampling and the Metropolis method are used to sample from the posterior distribution of the parameters. In each of the simulations, 200 samples were gathered after a burn-in of 1200 iterations. The MCMC was thinned by a factor of 5. (i.e. only one of every 5 samples are kept.)

Is this still true? I think that we discussed increasing the number of MCMC iterations to have more samples. Didn't we? Are you going to make a longer run?

4.3 Simulation Study I

A first simulation study was conducted to investigate how our model performs for simulated data (where the true value of the parameters are known) which has sample sizes and distributions similar to a real data-set (CB CYTOF data).

We simulate data so that

- N_i is on the order of 10000's
- μ^*_{zij} is reasonably far away from 0
- the latent feature matrix Z is "simple" (columns are linearly independent)
- the true number of latent features is 4

Please explain this in text, not in "itemize".

The MCMC is run for a sufficiently long time, and the dimensions of Z are fixed at 2,3,...,7. (i.e. six different models are run for different dimensions of Z.)

4.3.1 Simulated Data

Figure 1 show some of the properties of the simulated data. The number of rows in each of the matrices is in the order of tens of thousands. Specifically, $N = (20000, 30000, 10000)$. Note that these are heatmaps of y_{inj} (rather than $\tilde{y}_{inj}$). Red for positive values, blue for negative values, and white for missing values.

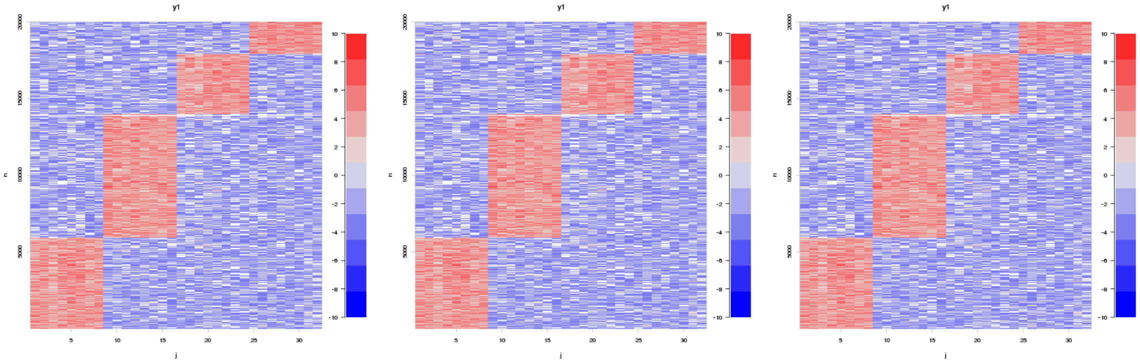


Figure 1: Simulated Data. Y_1 (left), Y_2 (middle), Y_3 (right).

`Figures 1: We rearranged` Y `according to their cell types. Please say it. Can we include missing values in the heatmap?`

In real CYTOF data, markers that are not expressed are sometimes recorded as having negative expression levels due to the mechanics of the measurement devices. Scientists interpret negative expression levels recorded by machines as non-expression. Consequently, we need to simulate data that have missing values. We do so by first generating data from the model, and then with some probability setting observations to be missing. Observations with lower values have a higher chance of becoming missing values. In this simulation, we record observations as missing using the function in Figure 2. The figure contains the simulation truth (in black) and the prior distribution over the logistic function (implied by the priors on β). Note that a strong prior is placed on logistic functions that are steep, and a loose prior is placed on the location of the logistic function.

4.3.2 Results

The parameters of greatest interest in this model are Z, W, and μ^*. Summaries of the posterior distributions from the simulations for those parameters will be discussed in this section.

4.3.2.1 Posterior Estimate of Z

In the current model, we assume the dimensions K of the latent feature allocation matrix to be known. In addition to understanding how well the model recovers Z when K is correctly

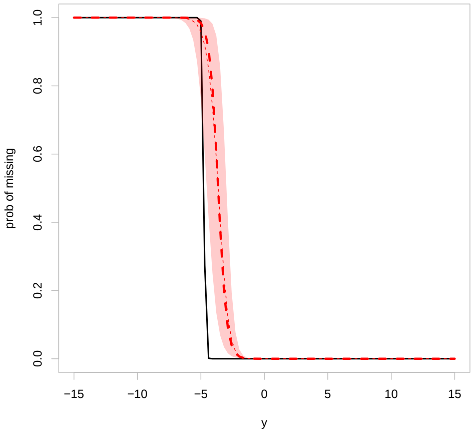


Figure 2: Probability of missing. The black line represents the simulation truth. The thick red dashed line represents the prior median. The thin red dashed line represents the prior mean. The red area is the prior 95% credible interval.

specified, we want to understand how mis-specifying the dimension (K) would affect the model. The following figures provide some insights to this objective. Recall that in the simulation truth, there are exactly four latent features. Figure 3 displays a simple latent feature matrix that was used in this simulation study.

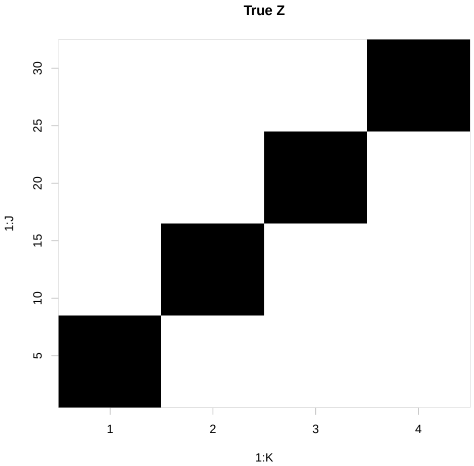


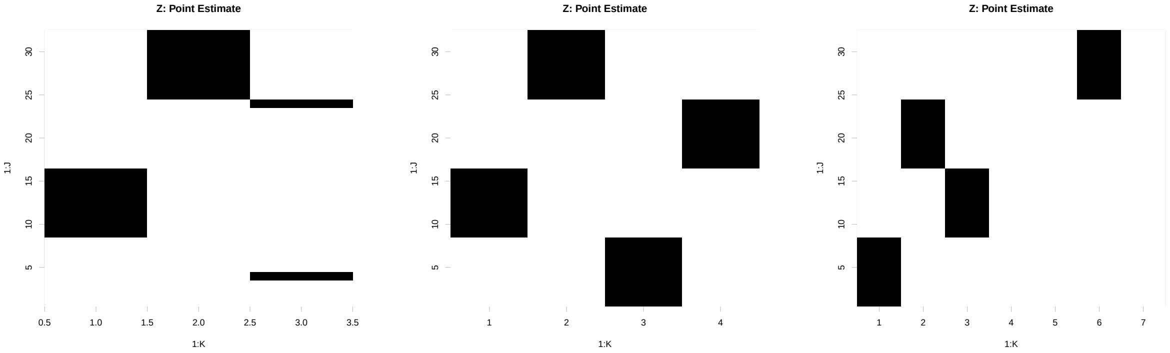
Figure 3: A simple Z matrix used for simulation study I.

I have summarized the posterior distribution of Z in two ways. The first is the posterior mean, which is simply the average of all posterior samples of Z from the MCMC. After averaging, the resulting matrix is sorted by column into its left-ordered form. The other summary statistic for Z is an adaptation of the sequentially-allocated latent structure optimization (SALSO) by David Dahl. In SALSO, a point estimate is obtained by finding a $\hat{Z}$ that minimizes the expression

$$\text{argmin}_Z \sum_{r=1}^{J}\sum_{c=1}^{J}(A(Z)_{rc} - \bar{A}_{rc})^2,$$

where $A(Z)$ is the pairwise allocation matrix corresponding to a binary matrix Z, and $\hat{A}$ is the pairwise allocation matrix averaged over all posterior samples of Z. The adaptation I have made is I have not used any optimization methods to compute $\hat{Z}$. I have simply selected the Z from the posterior samples of Z that minimizes the expression above. In this section, I'll mostly comment on the point-estimate of Z but I have included the posterior mean as a reference.

Figure 4 (left) shows the point-estimate for Z when the number of columns is fixed at 3 (mis-specified as smaller). In this case, the model learns 1 of the 4 columns of Z correctly. One column is duplicated. The remaining column shows no clear pattern. The effect is similar for when K is mis-specified as 2.



(a) $K = 3$ (b) $K = 4$ (c) $K = 7$

Figure 4: Posterior point-estimate for Z of 3 columns (left), 4 columns (middle), and 7 columns (right).

As we discussed, please remove "main titles" of the figures, make the texts on the axis bigger, make the texts on the axis informative (eg. the label for the y axis should be markers and 1.5 for k (cell types) does not make sense).

Figure 4 (middle) shows the point-estimate for Z when the number of columns is fixed at 4 (the truth). In this case, the true Z is learned.

Figure 4 (right) shows the point-estimate for Z when the number of columns is fixed at 7 (larger than the truth). In this case, the four columns of Z are learned correctly, two of the columns contain no activated features, and one column contains one active feature. This suggests that setting the dimensions of Z to be slightly higher may allow for the possibility of learning the correct structure for Z, at a slightly more computational cost. (Increasing the number of columns of Z in MCMC increases the log-computation time by a factor of $\log K$, while holding the sample-size constant).

4.3.2.2 Posterior Estimate of W

The W matrix, describes the proportion of observations within each sample that belong to a certain cell-type (one of K cell types in Z). The true W matrix used in the simulation studies is

$$W^{\text{TR}} = \begin{bmatrix} 0.3 & 0.4 & 0.2 & 0.1 \\ 0.1 & 0.7 & 0.1 & 0.1 \\ 0.2 & 0.3 & 0.3 & 0.2 \end{bmatrix}.$$

The interpretation of W_{ik} is the proportion of observations in sample i belonging to cell-type k.

The posterior mean of the W matrix for which $K = 3$ is

$$\hat{W}_3 = \begin{bmatrix} 0.40 & 0.10 & 0.50 \\ 0.70 & 0.10 & 0.20 \\ 0.30 & 0.20 & 0.50 \end{bmatrix}$$

The posterior mean is obtained by simple averaging of the posterior samples of W. Notice that in the posterior, since there are fewer columns of W than that in the truth, the proportions of the second column of W^{TR} are now in the first column of $\hat{W}_3$. The second column of $\hat{W}_3$ is close to 0. The remaining column takes the remaining proportions.

The posterior mean of the W matrix for which $K = 4$ is

$$\hat{W}_4 = \begin{bmatrix} 0.40 & 0.10 & 0.30 & 0.20 \\ 0.70 & 0.10 & 0.10 & 0.10 \\ 0.30 & 0.20 & 0.20 & 0.29 \end{bmatrix}$$

which closely resembles the truth (ignoring column ordering).

The posterior mean of the W matrix for which $K = 7$ is

$$\hat{W}_7 = \begin{bmatrix} 0.30 & 0.20 & 0.40 & 0.00 & 0.00 & 0.10 & 0.00 \\ 0.10 & 0.10 & 0.70 & 0.00 & 0.00 & 0.10 & 0.00 \\ 0.20 & 0.29 & 0.31 & 0.00 & 0.00 & 0.20 & 0.00 \end{bmatrix}$$

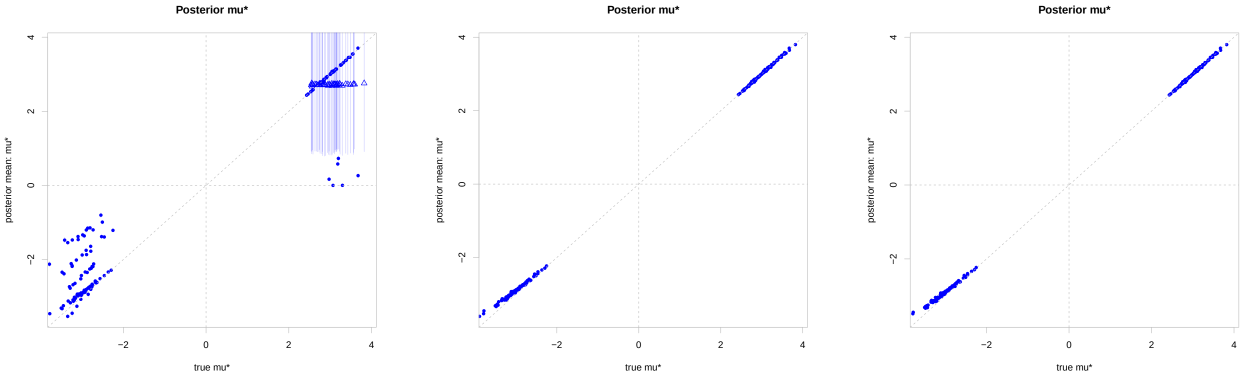
Ignoring the column ordering and the columns of zeros, $\hat{W}_7$ closely resembles the truth.

4.3.2.3 Posterior Estimate of μ^*

For example, in Figure **??** (left), we see a lot of uncertainty for μ^* which are supposed to be positively-valued. This is due to the posterior Z matrices lacking active features where they are needed (see Figure **??**).

In the case where Z is recovered correctly, μ^* can possibly be recovered correctly. Figure **??** (middle) shows the posterior distribution of μ^* for which $K = 4$. The posterior means line up with the truth. The credible intervals are short due to the number of observations.

Finally, Figure **??** (right) shows the posterior distribution of μ^* for which $K = 7$. As Z is recovered, the posterior means line up with the truth.



(a) $K = 3$ (b) $K = 4$ (c) $K = 7$

Figure 5: μ^* Posterior mean vs. true μ^* for $K = 3$ (left), $K = 4$ (middle), and $K = 7$ (right). Circles represent the posterior mean. Vertical lines represent the 95% credible intervals. Triangles also represent the posterior mean, but for μ_{zij} that have fewer than 30 corresponding $Z_{j,\lambda_{in}}$. They should have large intervals.