Clustering SNVs for Tumor Heterogeneity

Math-CS Sc.B Thesis

David Liu

January 31, 2017

Introduction

Cancer results from an evolutionary process where somatic mutations occur and accumulate in a population of cells. A mutation causes genetic variation at a genomic site called a single nucleotide variant (SNV) for the cell which obtained the mutation and all of its progeny. A mutation which causes selective advantages is called a driver mutation, which leads to this lineage of cells being in a greater proportion in a tumor. Other mutations may accumulate in this lineage but are selectively neutral; these mutations are called passenger mutations. The different lineages which comprise a tumor are known as clones, and the phenomenon of clonal admixture is known as intratumor heterogeneity.

(I plan on adding a figure here. Will add more background from the literature on tumor heterogeneity.)

Problem statement

One of the most important problems in tumor heterogeneity is tree inference, in which we estimate the evolutionary history and mixing proportions of clones. The first barrier to this problem is figuring out the mutations that belong to each clone—we must know which mutations correspond to a node before we can construct a tree from the nodes. This problem is made more tractable by incorporating multiple samples from the tumor, which provides more information for inference, since the clonal membership of SNVs is invariant across samples. We view the problem of assigning mutations to clones as a general machine learning problem of assigning mutations to clusters.

The data typically used for tree inference is called the variant allele frequency (VAF), observed as follows. Suppose we take multiple biopsy samples from a tumor separated spatially or temporally. These samples are sequenced separately, so that the reads are from a sample's mixture of clones. By comparing to a control sample, if a read contains the mutated allele at a SNV, it is called a variant read; otherwise it is called a reference read. The VAF is defined for each SNV in each sample, as, in each sample, the number of variant reads at an SNV divided by the number of total reads at that SNV. Each mutation which belongs to a clone should be observed to have about the same variant allele frequency, and this clustering should be true for each clone across all samples.

We can express the mathematical dependencies in our model in terms of the processes that generate them. First, each SNV $n = 1, ..., N$ must be assigned to a cluster $k = 1, \ldots, K$, $K < N$. These cluster memberships are described by the latent variables $\mathbf{z}_n$, a 1-of-K indicator vector that denotes the cluster assignment of SNV n to cluster k.

Now suppose that for each sample $m = 1, \ldots M$, we have total reads d_{mn} drawn from a Poisson with expected value of the coverage (Lander-Waterman cite). Let SNV n belong to cluster k. Then each cluster emits variant reads v_{mn} according to some distribution $V_{mk}(\boldsymbol{\phi}_{mk})$, where $\boldsymbol{\phi}_{mk}$ are the parameters for V_{mk}. We can vectorize this as

$$\mathbf{v_n} = \begin{bmatrix} v_{1n} \\ v_{2n} \\ \vdots \\ v_{Mn} \end{bmatrix} \sim \begin{bmatrix} V(d_{1n}, \boldsymbol{\phi}_{1k}) \\ V(d_{2n}, \boldsymbol{\phi}_{2k}) \\ \vdots \\ V(d_{Mn}, \boldsymbol{\phi}_{Mk}) \end{bmatrix} = \mathbf{V}(\boldsymbol{\phi}_\mathbf{k}) \tag{1}$$

Let $\mathbf{x}_n$ be general notation for $\{d_n, v_n\}$, where the use of the reference or variant reads will be clear from context. Then we wish to discover the underlying $\mathbf{z}_n, \boldsymbol{\phi}$ for the model given some observations $\mathbf{x}$.

There are existing clustering methods in bioinformatics such as PyClone and SciClone (cite). These methods choose V_{mn} to be a binomial distribution, or choose V_{mn} to be beta with data $f_{mn} = \frac{v_{mn}}{d_{mn}}$. While they both can use a multi-binomial mixture model, their model selection for the number of clusters is through a Dirichlet prior and an ad-hoc heuristic. In terms of inference, PyClone uses MCMC to approximate the posterior while SciClone uses variational inference. MCMC, while accurate in the long run, may have poor convergence properties, while variational inference is a faster technique that potentially trades off some accuracy for speed and scalability.

However, state of the art clustering methods use the Dirichlet process to select the number of clusters, which as a nonparametric model, has more rigorous model selection when compared to the heuristic methods used in PyClone and SciClone. There is also a need for inference on large datasets, for which variational inference is useful. In this thesis, I address this need by proposing and implementing a method to cluster mutations using variational inference for a multi-binomial mixture model with Dirichlet process prior, which is suited for the multi-sample clone mixing problem.

Multi-sample Binomial Mixture Model with DP prior

This section follow the notation introduced above.

Because reads follow a reference/variant pattern, suppose that each clone in each sample emits variant reads according to a binomial distribution—thus we choose V_{mn} to be a binomial distribution. Thus, for cluster k, some cluster member n, and reads d_{mn}, we have variant reads distributed according to $\mathrm{Bin}(v_{mn}; d_{mn}, \boldsymbol{\phi}_{mk})$. We can consider the joint probability of reads for an SNV by the product across all samples, since we assume samples are independent. That is,

$$p(\mathbf{x_n}|\boldsymbol{\phi}_k) = \prod_{m=1}^{M} \mathrm{Bin}(v_{mn}; d_{mn}, \boldsymbol{\phi}_k) \tag{2}$$

Further suppose that the cluster memberships $\mathbf{z}_n$ and weights π_k are generated by a Dirichlet process prior. The reader is referred to (Antoniak 1974) for more mathematical detail on the DP. Thus K could be countably infinite, but with probability one, K is finite.

As we can see by inspecting the graphical model, the likelihood of $\mathbf{x}$ depends on the latent variables in a straightforward way:

$$\begin{aligned} p(\mathbf{x_n}|\mathbf{z}, \boldsymbol{\phi}_k) &= \prod_{k=1}^{K} p(\mathbf{x_n}|\phi_{\mathbf{k}})^{\mathbf{z}_{nk}} \\ &= \prod_{k=1}^{K} \prod_{m=1}^{M} \mathrm{Bin}(v_{mn}; d_{mn}, \boldsymbol{\phi}_{mk})^{\mathbf{z}_{nk}} \end{aligned} \tag{3}$$

Now consider the joint likelihood of the observed data and latent variables, which follows from (3):

$$p(\mathbf{x_n}, \mathbf{z}|\boldsymbol{\pi}, \boldsymbol{\phi}) = \prod_{k=1}^{K} \prod_{m=1}^{M} \left(\pi_k \mathrm{Bin}(v_{mn}; d_{mn}, \phi_{mk})\right)^{\mathbf{z}_{nk}} \tag{4}$$

Figure 1 on the next page shows the graphical model.

$n = 1, \ldots, N$: SNVs
$m = 1, \ldots, M$: samples
$k = 1, \ldots, K$: clusters

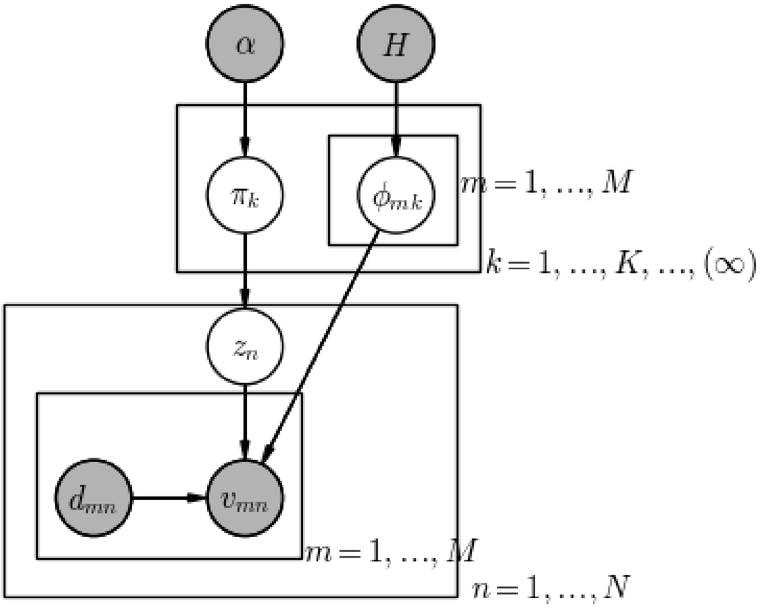

Figure 1: Graphical model for the VAFs.

α = Hyperparameter for the stick-breaking process
$H \sim U(0,1) \sim \text{Beta}(1,1)$ = Base distribution for parameters ϕ_{mk}
π_k = Cluster weights, generated from the stick-breaking process
$z_n \in \{1, \ldots, K, \ldots\} \sim \text{Cat}_\infty(\pi_1, \ldots, \pi_K, \ldots)$ = Cluster membership for SNV n
ϕ_{mk} = Cluster frequency
$v_{mn} \sim \text{Binom}(v_{mn}; d_{mn}, \phi_{mk})$ = Observed variant reads for sample m, SNV n, belonging to cluster k.
$d_{mn} \sim \text{Pois}(\text{Coverage})$ = Observed total reads for sample m, SNV n

As described in (Blei 2006), the data arises in the following manner, with a stick-breaking construction given as:

$$\pi_i(\mathbf{v}) = v_i \prod_{j=1}^{i-1} (1 - v_j)$$

$$DP = \sum_{i=1}^{\infty} \pi_i(\mathbf{v}) \delta_{\phi_i}$$

1. Draw $v_k | \alpha \sim \text{Beta}(1, \alpha), \quad k = \{1, 2, \ldots\}$
2. Draw $\boldsymbol{\phi}_k | H \sim H^M, \quad k = \{1, 2, \ldots\}$
3. For the nth data point:
   (a) Draw $z_n | \{\pi_1, \pi_2, \ldots\} \sim \text{Cat}(\pi(\mathbf{v}))$.
   (b) Draw $\mathbf{x}_n | z_n \sim p(\mathbf{x}_n | \boldsymbol{\phi}_k)$.

The full posterior

$$p(\mathbf{z}|\mathbf{x}, \alpha, H) = \int p(\mathbf{x}|\phi) p(\phi|\mathbf{x}, \alpha, H)\, d\phi \tag{5}$$

involves a Dirichlet Process and is thus analytically intractable. We must use some sort of computational technique to perform inference on this posterior.

Variational Inference

Variational inference is an alternative to MCMC-based inference methods. At a high level, variational inference factors a posterior using the mean-field approximation, which approximates the posterior in a higher-dimensional space using simpler independent functions. Then a simple coordinate ascent can be performed in order to infer the model parameters.

The ELBO

Let $\mathbf{z}$ denote the latent variables, and $\mathbf{x}$ denote the data. We seek to approximate the posterior $p(\mathbf{z}|\mathbf{x})$ from a family of distributions $\mathcal{D}$ by solving the following optimization problem:

$$q^*(z) = \arg\min_{q(\mathbf{z}) \in \mathcal{D}} \textsc{kl}\left((q(\mathbf{z})||p(\mathbf{z}|\mathbf{x}))\right). \tag{6}$$

where KL is the KL-divergence, which measures the “distance” between two distributions.

However, (6) requires us to compute the log evidence (which is intractable over the space of all $\mathbf{z}$) since

$$\textsc{kl}\left(q(\mathbf{z}) \| p(\mathbf{z} \mid \mathbf{x})\right) = \mathrm{E}\left[\log q(\mathbf{z})\right] - \mathrm{E}\left[\log p(\mathbf{z}, \mathbf{x})\right] + \log p(\mathbf{x}). \tag{7}$$

Instead, we optimize an objective function which is not dependent on $\log p(\mathbf{x})$. We call this the evidence lower bound (ELBO), which is equal to the negative KL-divergence plus the log evidence.

$$\textsc{elbo}(q) = \mathrm{E}[\log p(\mathbf{z}, \mathbf{x})] - \mathrm{E}[\log q(\mathbf{z})]. \tag{8}$$

and thus we see that $\log p(\mathbf{x})$ is a constant with respect to q. The ELBO gets its name from the fact that it is a lower bound for the log evidence.

The mean-field variational family

And what family of distributions do we use for $\mathcal{D}$? The standard technique is to use a simple one from physics, the mean-field variational family. In this family, the latent variables $\mathbf{z}$ are mutually independent so that the joint distribution factorizes:

$$q(\mathbf{z}) = \prod_{j=1}^{m} q_j(z_j). \tag{9}$$

where q_j is a bounded variation dependent only on z_j. The structure of the model will dictate the optimal form of q_j.

Coordinate ascent

The optimization is solved using a coordinate ascent algorithm, where via the mean-field assumption, the independence of the latent variables gives us orthogonality. Let z_{-j} denote the set of latent variables z_l such that $l \neq j$. Consider the complete conditional of z_j, which is a function of the other latent variables and the data, $p(z_j \,|\, \mathbf{z}_{-j}, \mathbf{x})$. Since the expectation in the ELBO is with respect to $q(\mathbf{z})$, which we have assumed factorizes, then we can dissect out the dependence with respect to $\mathbf{z}_j$ by using (8) and (9):

$$\begin{aligned}\text{ELBO}(q) &= \int \prod q_i(\mathbf{z}_i) \left(\log p(\mathbf{z}, \mathbf{x}) - \sum_i \log q_i(\mathbf{z}_i) \right) d\mathbf{z} \\ &\propto \int q_j(z_j) \mathrm{E}_{-j} \left[\log p(\mathbf{x}, \mathbf{z}) \right] \, d\mathbf{z}_j - \int q_j(z_j) \log q_j(\mathbf{z}_j) \, d\mathbf{z}_j \qquad (10)\end{aligned}$$

Now suppose that we fix z_{-j} and maximize the ELBO. Then the ELBO is maximized when $\log q_j(\mathbf{z}_j) \propto \mathrm{E}_{-j} \left[\log p(\mathbf{x}, \mathbf{z}) \right]$, by the positivity of the KL-divergence. Thus the optimal $q^*(\mathbf{z}_j)$ occurs when

$$q_j^*(\mathbf{z}_j) \propto \exp\left(\mathrm{E}_{-j} \left[\log p(\mathbf{x}, \mathbf{z}) \right] \right) \qquad (11)$$

(11) underlies the coordinate-ascent variational inference algorithm. By iterating through each variational factor, fixing the others, and performing coordinate ascent (similar to Gibbs sampling), then we eventually reach a local optimum of the ELBO.

Exponential family distributions yield a general formula

If our posterior is in the exponential family, then the computation of coordinate ascent and ELBO can be generalized. Recall that a distribution is in exponential form if it can parameterized by

$$f_X(x \mid \theta) = h(x) \exp\left(\theta^T \cdot T(x) - A(\theta) \right)$$

where $T(x)$ is the sufficient statistic vector, θ is the natural parameter vector, and $A(\theta)$ is the cumulant. We keep the actual derivations to (Hughes 2015), but the intuition is that because the optimal variational updates are proportional to a exp(E[log(.)]) then writing the distribution in exponential form reveals some dependencies that hold for all exponential family members.

Variational Inference on Multi-sample Binomial Model

The model and its ELBO

We write the ELBO as a function of the data and latent variables:

$$\begin{aligned}\text{ELBO}\left(q(\mathbf{x}, \mathbf{z} | \gamma, \alpha_0, \beta_0) \right) = & E_q[\log p(\mathbf{v}|\gamma)] + E_q[\log p(\boldsymbol{\phi}|\alpha_0, \beta_0)] + \\ & \sum_{n=1}^{N} \left(E_q[\log p(z_n|\mathbf{v})] + E_q[\log p(x_n|z_n)] \right) \qquad (12) \\ & - E_q[\log q(\mathbf{z}, \mathbf{v}, \boldsymbol{\phi})]\end{aligned}$$

where λ represents the hyperparameter governing the stick-breaking process, and α_0, β_0 are the hyperparameters governing the base beta distribution. By the mean-field assumption, the joint distribution for the last term in the ELBO factors as follows:

$$q(\mathbf{z}, \mathbf{v}, \boldsymbol{\phi}) = \underbrace{\prod_{k=1}^{K} q(\boldsymbol{\phi}_k)}_{\substack{\text{Observation: likelihoods}\\ \text{Product of betas}\\ 2MK \text{ variational parameters}\\ \{\alpha_{mk},\beta_{mk}\}_{m=1,k=1}^{M,K}}} \times \underbrace{\prod_{k=1}^{K} q(\mathbf{v}_k)}_{\substack{\text{Allocation: cluster proportions}\\ \text{Product of betas}\\ 2K \text{ variational parameters}\\ \{\eta_{k0},\eta_{k1}\}_{k=1}^{K}}} \times \underbrace{\prod_{n=1}^{N} q(z_n)}_{\substack{\text{Allocation: cluster responsibilities}\\ \text{Product of categoricals}\\ 2NK \text{ variational parameters}\\ \{\hat{r}_{nk}\}_{n=1,k=1}^{N,K}}}$$

Note that the cluster proportions and data assignments are dictated by the Dirichlet Process—we call this the allocation model. On the other hand, the observation parameters vary depending on the structure of the generative model—we call this the observation model. The equations relating to the allocation model are standard, and the theory is left to (Blei 2006, Hughes 2015). Here we derive the form of the $q(\boldsymbol{\phi}_k)$ is a function of $\boldsymbol{\phi_k}$, as the observation model is specific to our multi-sample binomial model.

We note that for an individual allelic site in a sample, the data likelihood is binomial. With a beta prior, we know that the resulting posterior for $q(\boldsymbol{\phi}_{mk})$ is conjugate to the binomial, and thus $q(\boldsymbol{\phi}_{mk}) \sim \text{Beta}(\boldsymbol{\phi}_k|\alpha_{mk}, \beta_{mk})$ where α_{mk}, β_{mk} are variational parameters. Because reads across samples at a site are assumed to be independent, then we have

$$\begin{aligned} q(\boldsymbol{\phi}_k) &= \prod_{m=1}^{M} q(\boldsymbol{\phi}_{mk}) \\ &= \prod_{m=1}^{M} \text{Beta}(\boldsymbol{\phi}_k|\alpha_{mk}, \beta_{mk}) \end{aligned}$$

Coordinate ascent algorithm

The details of the derivations of the coordinate ascent algorithm are left to Appendix A. The details of the derivations for the ELBO are left to Appendix B. Taking results from these two appendices, we have the following procedure for coordinate ascent on our model.

Initialization

The initial responsibilities of the cluster were chosen by setting $\hat{r}_{nk} = 1$ if n was set to be in cluster k by the k-means++ algorithm with $\frac{N}{2}$ initial clusters, with the other $\hat{r}_{n.}$ set to be $\frac{1}{k}$. These responsibilities were then normalized. For the other parameters, we assume that they are set to their prior or uniform values. The truncation level of K was set to be $\frac{N}{2}$, with all K greater than $\frac{N}{2}$ being irrelevant.

Convergence

We declare the coordinate ascent procedure to be complete when the difference in ELBO between two iterations is less than some convergence threshold. Empirically, we chose the threshold to be equal to 0.01.

```
Algorithm 1: CAVI FOR THE MULTIDIMENSIONAL BINOMIAL MODEL
Input: Data $\mathbf{x}_n$, where each $x_i$ is an integer vector with $M$ entries.
       $\gamma_0, \gamma_1, \alpha_0, \beta_0$, hyperparameters
Output: Converged variational parameters $\{\alpha_{mk}, \beta_{mk}\}_{m=1,k=1}^{M,K}, \{\eta_{k0}, \eta_{k1}\}_{k=1}^{K}, \{\hat{r}_{nk}\}_{n=1,k=1}^{N,K}$
Initialize: $\alpha_0 = \beta_0 = \alpha_{mk} = \beta_{mk} = 1, \forall m, k$
            $\gamma_1 = \eta_1 = 1.0, \gamma_0 = \eta_0 = 1.5$
            $\hat{r}_{nk} \leftarrow \texttt{kmeans++}(\mathbf{x})$
while the ELBO has not converged do
    ▷ Compute data-specific (local) parameters
        $\mathrm{E}_q[\log p(x_n|\alpha_{mk}, \beta_{mk})] \leftarrow \mathrm{E}_q[\log \left(\binom{d_{mn}+v_{mn}}{v_{mn}}(\boldsymbol{\phi}_k)^{v_{mn}}(1-\boldsymbol{\phi}_k)^{d_{mn}}\right)]$
        $\hat{r}_{nk} \leftarrow \exp(S_k)$
    ▷ Compute sufficient statistics
        $S_k = \sum_{n=1}^{N} \hat{r}_{nk} s(x_n) = \sum_{n=1}^{N} \hat{r}_{nk} \left[\begin{bmatrix} v_{1n} & d_{1n} \end{bmatrix} \cdots \begin{bmatrix} v_{Mn} & d_{Mn} \end{bmatrix}\right]$
        $N_k = \sum_{n=1}^{N} \hat{r}_{nk}$
        $N_k^{>} = \sum_{k+1}^{K} N_k$
    ▷ Compute cluster-specific (global) parameters
        $\eta_{k1} \leftarrow 1 + \sum_n \hat{r}_{nk} = 1 + N_k$
        $\eta_{k0} \leftarrow \gamma + \sum_n \sum_{j=k+1}^{K} \hat{r}_{nj} = N_k^{>}$
        $\alpha_{mk} \leftarrow (\alpha_0 - 1) + S_{km}$
        $\beta_{mk} \leftarrow (\beta_0 - 1) + S_{km}$
    Compute $\mathrm{ELBO}(q) = \mathbb{E}[\log p(\mathbf{z}, \mathbf{x})] + \mathbb{E}[\log q(\mathbf{z})]$
end
return Converged variational parameters
```

MAP estimates

We make MAP estimates by converting from variational parameters back to the original parameters of the posterior:

$$\mathbf{z}_n = \arg\max_k \hat{r}_{nk} \tag{13}$$

$$\phi_{mk} = \frac{v^{mk} + \alpha_{mk} - 1}{d^{mk} + \alpha_{mk} + \beta_{mk} - 2} \tag{14}$$

where we mean by $\mathbf{z}_n = k$ that $\mathbf{z}_n$ is a 1-of-k indicator vector. And by pooling reads,

$$v^{mk} = \sum_n (v_{mn})^{\mathbf{z}_n} \tag{15}$$

$$d^{mk} = \sum_n (d_{mn})^{\mathbf{z}_n} \tag{16}$$

Implementation

This has been implemented in Python.

Experiments and Results

The “positive control” should be the SciClone model, when it is given the correct number of clusters.

Comparison against other clustering methods.

- SciClone
- Gaussian DP
- Ancestree SCC

→ Real data

Future work

- Add to bnpy to get stochastic/memoized VI
- Use the results as input for other algorithms.

Appendix A: Deriving update equations

Allocation model

The coordinate ascent equations for the allocation model are standard (Blei 2006), as they follow from the fact that the stick-breaking process is in the exponential family. The allocation model has variational parameters $\{\eta_{k0}, \eta_{k1}\}_{k=1}^{K}$ for the cluster proportions and $\{\hat{r}_{nk}\}_{n=1,k=1}^{N,K}$ for the cluster responsibilities. On each iteration, the coordinate update is

$$\eta_{k1} = 1 + \sum_{n} \hat{r}_{nk} = 1 + N_k \tag{17}$$

$$\eta_{k0} = \gamma + \sum_{n} \sum_{j=k+1}^{K} \hat{r}_{nj} = N_k^{>} \tag{18}$$

$$\hat{r}_{nk} \propto \exp(S_k) \tag{19}$$

for $n = 1, \ldots, N$, $k = 1, \ldots, K$, and where

$$S_k = \mathrm{E}_q[\log \boldsymbol{v}_k] + \sum_{i=1}^{k-1} \mathrm{E}_q \log(1 - \boldsymbol{v}_i) + \mathrm{E}_q[\log p(x_n|\alpha_{nk}, \beta_{nk})] \tag{20}$$

and

$$
\begin{aligned}
\mathrm{E}_q[\log \boldsymbol{v}_i] &= \Psi(\eta_{k0}) - \Psi(\eta_{k0} + \eta_{k1}) && (21)\\
\mathrm{E}_q[\log(1 - \boldsymbol{v}_i)] &= \Psi(\eta_{k1}) - \Psi(\eta_{k0} + \eta_{k1}) && (22)
\end{aligned}
$$

The digamma functions come from the fact that derivative of the cumulant is the expectation, and the cumulant of a beta has gamma functions (http://math.stackexchange.com/questions/1603172/digamma-function-in-expectation). Since the $\hat{r}_{nk}$ sum to 1 over $n = 1, \ldots, N$ then we renormalize at every step as well.

Observation model

Following the derivations in (Hughes 2015), we derive the coordinate ascent equations for our observation model, taking advantage of the fact that $q(\boldsymbol{\phi}_k)$ is in the exponential family, which we show below.

Claim 1. *$q(\boldsymbol{\phi}_k)$ is in the exponential family.*

Proof. We know that

$$
q(\boldsymbol{\phi}_k) = \prod_{m=1}^{M} q(\boldsymbol{\phi}_{mk}) = \prod_{m=1}^{M} \mathrm{Beta}(\boldsymbol{\phi}_k | \alpha_{mk}, \beta_{mk}).
$$

The beta distribution is in the exponential family, with parameterization

$$
\begin{aligned}
\mathrm{Beta}(\boldsymbol{\phi}_k | \alpha_{mk}, \beta_{mk}) = \frac{1}{\phi_{mk}(1 - \phi_{mk})} \exp\Bigg(&\begin{bmatrix} \log \phi_{mk} & \log(1 - \phi_{mk}) \end{bmatrix} \begin{bmatrix} \alpha_{mk} \\ \beta_{mk} \end{bmatrix} \\
&+ \log \Gamma(\alpha_{mk} + \beta_{mk}) - \log \Gamma(\alpha_{mk}) - \log \Gamma(\beta_{mk}) \Bigg)
\end{aligned} \tag{23}
$$

and thus $q(\boldsymbol{\phi}_k)$ is in the exponential family with the form

$$
\begin{aligned}
q(\boldsymbol{\phi}_k) = \left(\prod_{m=1}^{M} \frac{1}{\phi_{mk}(1 - \phi_{mk})} \right) \exp\Bigg(&\begin{bmatrix} \begin{bmatrix} \log \phi_{1k} & \log(1 - \phi_{1k}) \end{bmatrix} & \cdots & \begin{bmatrix} \log \phi_{Mk} & \log(1 - \phi_{Mk}) \end{bmatrix} \end{bmatrix} \begin{bmatrix} \begin{bmatrix} \alpha_{1k} \\ \beta_{1k} \end{bmatrix} \\ \vdots \\ \begin{bmatrix} \alpha_{Mk} \\ \beta_{Mk} \end{bmatrix} \end{bmatrix} \\
&+ \sum_{m=1}^{M} \log \Gamma(\alpha_{mk} + \beta_{mk}) - \log \Gamma(\alpha_{mk}) - \log \Gamma(\beta_{mk}) \Bigg)
\end{aligned}
$$

where we have abused notation to show the tuple nature of the sufficient statistics. Thus, for our variational distribution $q(\boldsymbol{\phi}_k)$ we have

$$\text{Natural parameters} = \begin{bmatrix} \begin{bmatrix} \alpha_{1k} \\ \beta_{1k} \end{bmatrix} \\ \vdots \\ \begin{bmatrix} \alpha_{Mk} \\ \beta_{Mk} \end{bmatrix} \end{bmatrix} \tag{24}$$

$$\text{Cumulant} = \sum_{m=1}^{M} \log\Gamma(\alpha_{mk} + \beta_{mk}) - \log\Gamma(\alpha_{mk}) - \log\Gamma(\beta_{mk}) \quad \Box \tag{25}$$

Exponential factorization of data model

$$\begin{aligned} p(\mathbf{x}_n|\mathbf{z}_n) &= \prod_{m=1}^{M} \text{Bin}(v_{mn}; \phi_{mn}, d_{mn}) \\ &= \prod_{m=1}^{M} \exp\left(v_{mn} \log\left(\frac{\phi_{mn}}{1-\phi_{mn}} \right) + (d_{mn} - v_{mn}) \log(1-\phi_{mn}) \right) \\ &= \left(\prod_{m=1}^{M} \binom{d_{mn}}{v_{mn}} \right) \exp\left(\begin{bmatrix} \log\left(\frac{\phi_{1n}}{1-\phi_{1n}}\right) & \cdots & \log\left(\frac{\phi_{Mn}}{1-\phi_{Mn}}\right) \end{bmatrix} \begin{bmatrix} v_{1n} \\ \vdots \\ v_{Mn} \end{bmatrix} + \right. \\ &\qquad \left. \begin{bmatrix} \log(1-\phi_{1n}) & \cdots & \log(1-\phi_{Mn}) \end{bmatrix} \begin{bmatrix} d_{1n} \\ \vdots \\ d_{Mn} \end{bmatrix} \right) \qquad (26) \\ &= \left(\prod_{m=1}^{M} \binom{d_{mn}}{v_{mn}} \right) \exp\left(\begin{bmatrix} \begin{bmatrix} v_{1n} & d_{1n} \end{bmatrix} & \cdots & \begin{bmatrix} v_{Mn} & d_{Mn} \end{bmatrix} \end{bmatrix} \begin{bmatrix} \begin{bmatrix} \log(1-\phi_{1n}) \\ \log\left(\frac{\phi_{1n}}{1-\phi_{1n}}\right) \end{bmatrix} \\ \vdots \\ \begin{bmatrix} \log(1-\phi_{Mn}) \\ \log\left(\frac{\phi_{Mn}}{1-\phi_{Mn}}\right) \end{bmatrix} \end{bmatrix} \qquad (27) \\ & \qquad (28) \end{aligned}$$

so that the sufficient statistics

$$T(\mathbf{x}_n) = \begin{bmatrix} \begin{bmatrix} v_{1n} & d_{1n} \end{bmatrix} & \cdots & \begin{bmatrix} v_{Mn} & d_{Mn} \end{bmatrix} \end{bmatrix} \tag{29}$$

Thus, following (Hughes 2015) we have the following coordinate ascent updates for the observation model:

(natural parameter plus sufficient statistic S_k^{var} or S_k^{ref})

$$\alpha_{mk} = (\alpha_0 - 1) + \sum_{n=1}^{N} \hat{r}_{nk} \begin{bmatrix} v_{1n} \\ \vdots \\ v_{Mn} \end{bmatrix}$$

$$\beta_{mk} = (\beta_0 - 1) + \sum_{n=1}^{N} \hat{r}_{nk} \begin{bmatrix} d_{1n} \\ \vdots \\ d_{Mn} \end{bmatrix}$$

Sufficient statistics

Define

$$S_k = \sum_{n=1}^{N} \hat{r}_{nk} s(x_n) = \sum_{n=1}^{N} \hat{r}_{nk} \left[\begin{bmatrix} v_{1n} & d_{1n} \end{bmatrix} \quad \cdots \quad \begin{bmatrix} v_{Mn} & d_{Mn} \end{bmatrix} \right] \tag{30}$$

$$N_k = \sum_{n=1}^{N} \hat{r}_{nk} \tag{31}$$

$$N_k^{>} = \sum_{k+1}^{K} N_k \tag{32}$$

Obs Model Likelihoods

$$\begin{aligned} \mathrm{E}_q[\log p(x_n|\alpha_{mk}, \beta_{mk})] &= \mathrm{E}_q[\log \left(\binom{d_{mn} + v_{mn}}{v_{mn}} (\boldsymbol{\phi}_k)^{v_{mn}} (1 - \boldsymbol{\phi}_k)^{d_{mn}} \right)] \\ &= \log \left(\binom{d_{mn} + v_{mn}}{v_{mn}} \right) + d_{mn} \mathrm{E}_q[\log \boldsymbol{\phi}_k] + v_{mn} \mathrm{E}_q[\log(1 - \boldsymbol{\phi}_k)] \end{aligned}$$

where

$$\begin{aligned} \mathrm{E}_q[\log \boldsymbol{\phi}_k] &= \Psi(\alpha_{mk}) - \Psi(\alpha_{mk} + \beta_{mk}) \\ \mathrm{E}_q[\log(1 - \boldsymbol{\phi}_k)] &= \Psi(\beta_{mk}) - \Psi(\alpha_{mk} + \beta_{mk}) \end{aligned}$$

Appendix: Computing the ELBO

To test for convergence, we calculate the ELBO until the difference in ELBO between laps is less than some pre-specified number. For the purpose of this model, the convergence threshold was set to 1. Note that the ELBO is generally not a convex function, so we cannot make any guarantees about monotonicity.

Following (Hughes 2015), the ELBO can be decomposed into three terms:

$$\text{ELBO} \coloneqq \mathcal{L} = \mathcal{L}_{\text{Obs}} + \mathcal{L}_{\text{DP-Alloc}} + \mathcal{L}_{\text{Entropy}}$$

Observation model contribution to ELBO

$$
\begin{aligned}
\mathcal{L}_{\text{Obs}} &= \mathrm{E}_{\mathbf{z},\boldsymbol{\phi}}[\log p(x|\mathbf{z},\boldsymbol{\phi})] + \mathrm{E}_{\phi}[\log p(\boldsymbol{\phi})] - \mathrm{E}_{\phi}[\log q(\boldsymbol{\phi})] \qquad (33) \\
&= \sum_{n=1}^{N}\sum_{k=1}^{K} \hat{r}_{nk}\mathrm{E}_q[\log p(\mathbf{x}_n|\boldsymbol{\phi}_k)] \\
&\quad + \sum_{k=1}^{K}\sum_{m=1}^{M} \mathrm{E}_q[\log \boldsymbol{\phi}_k^0] \qquad (34) \\
&\quad - \sum_{k=1}^{K}\sum_{m=1}^{M} \mathrm{E}_q[\log \boldsymbol{\phi}_k]
\end{aligned}
$$

Allocation model contribution to ELBO

$$
\begin{aligned}
\mathcal{L}_{\text{DP-Alloc}} = &\sum_{k=1}^{K} c_{\text{Beta}}(1,\gamma) - c_{\text{Beta}}(\eta_{k1},\eta_{k0}) \\
&+ \sum_{k=1}^{K} (N_k + 1 - \eta_{k1})\mathrm{E}_q[\log \boldsymbol{u}_k]) \qquad (35) \\
&= \sum_{k=1}^{K} \left(N_k^{>} + \gamma - \eta_{k0}\right) \mathrm{E}_q[\log(1-\boldsymbol{u}_k)]
\end{aligned}
$$

where the expectations are defined above, and in (18), we showed that c_{Beta} has the form

$$
c_{\text{Beta}}(\alpha,\beta) = \log\Gamma(\alpha+\beta) - \log\Gamma(\alpha) - \log\Gamma(\beta) \qquad (36)
$$

Entropy contribution to ELBO

$$
\mathcal{L}_{\text{Entropy}} = -\sum_{k=1}^{K}\sum_{n=1}^{N} \hat{r}_{nk}\log\hat{r}_{nk}
$$