Chapter 5

Parallel Split-Merge MCMC for the DPMM

COMPUTER vision problems are often difficult because of the sheer size of data to process. As such, developments in the probabilistic modeling community are often impeded in applications to computer vision. We believe one such development is in the Bayesian nonparametric models (e.g., Dirichlet process mixture models and their extensions), where inference that can handle the large number of observations in computer vision problems are still lacking. Motivated by this observation, we spend the next two chapters developing scalable inference algorithms for Dirichlet process mixture models (DPMMs) and their extensions. This work is a slight departure from the previous focus on computer vision and is a standalone contribution to the machine learning and probabilistic modeling community. In Chapter 7, we apply the developed inference algorithms can to the classic computer vision of intrinsic image decomposition.

Mixture models are a commonly used framework to model clusters of data in the machine learning community. In the recent decades, there has been considerable interest in extending classical finite mixture models to exploit non-parametric Bayesian statistics. Among other things, the elegant theory behind Dirichlet process mixture models (DPMM) has extended finite mixture models to include automatic model selection in clustering problems.

However, the rich representative power of the DP comes at a cost; unlike finite mixture models where the number of components is known *a priori* and can be fully instantiated, the infinite number of components cannot be represented. This has led to much work on developing methods of posterior inference, some of which are summarized in Section 2.9.2. One common approach is to perform Markov chain Monte Carlo sampling with Gibbs sampling, but this often leads to undesirable results because samplers that propose local changes exhibit poor convergence. Split and merge moves, first considered in DPs by [63], attempt to address these convergence issues, but currently cannot be parallelized and often waste precious computation on proposing a split or merge that is simply rejected by a Metropolis-Hastings ratio. Little work has been done in developing *scalable* split/merge moves for large datasets. Alternatively, approximate inference based on asymptotic limits such as [73] or variational approx-

Table 5.1: Capabilities of MCMC Sampling Algorithms in DPMMs

	CW	[60, 61]	[32, 96]	[27, 45, 63]	[64]	[83, 127]	Proposed Method
Exact Model	✓	·	✓	✓	✓	✓	✓
Splits & Merges	·	·	·	✓	✓	·	✓
Intra-cluster Parallel	·	·	·	·	·	✓	✓
Inter-cluster Parallel	·	✓	✓	·	·	·	✓
Non-conjugate Priors	✓	✓	✓	·	✓	·	✓

imations such as [10] can be used. Small-variance asymptotics perform inference on an altered model and variational algorithms do not have the limiting guarantees of MCMC methods. Both such methods may also suffer from similar convergence issues, but are particularly appealing for use in large datasets as they often lend themselves to parallelization.

In this chapter, we develop an MCMC sampling technique for mixture models that: (1) preserves limiting guarantees; (2) proposes splits and merges to improve convergence; (3) parallelizes and scales well for use in large datasets; and (4) is applicable to a conjugate and non-conjugate priors. To our knowledge, no current sampling algorithms satisfy all of these properties simultaneously. While this chapter mainly focuses on Dirichlet process mixture models, we note that similar methods can be applied for mixture models with other priors (finite Dirichlet distributions, Pitman-Yor Processes, etc.). An earlier version of this work was originally presented in [21].

■ 5.1 Related Work

We briefly review relevant related work in MCMC sampling for the Dirichlet process mixture model. The majority of DPMM samplers fit into one of two categories: collapsed-weight samplers that marginalize over the mixture weights or instantiated-weight samplers that explicitly represent them. Capabilities of current algorithms, which we now overview, are summarized in Table 5.1. A more detailed description of some related work is given in Section 2.9.2.

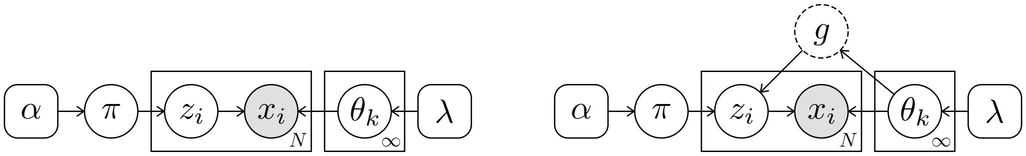
Collapsed-weight (CW) samplers using both conjugate (e.g. [16, 30, 85, 92, 126]) and non-conjugate (e.g. [86, 93]) priors sample the cluster labels iteratively one data point at a time without needing to approximate the infinite-length model. When a conjugate prior is used, one can also marginalize out cluster parameters. However, as noted by multiple authors (e.g. [27, 63, 79]), these methods often exhibit slow convergence. Additionally, due to the particular marginalization schemes, these samplers cannot be parallelized.

Instantiated-weight (IW) samplers explicitly represent cluster weights, typically using a finite approximation to the DP (e.g. [60, 61]). Recently, [32] and [96] have eliminated the need for this approximation; however, IW samplers still suffer from convergence issues. If cluster parameters are marginalized, it can be very unlikely for a single point to start a new cluster. When cluster parameters are instantiated, samples of parameters from the prior are often a poor fit to the data. However, IW samplers are often useful because they can be parallelized across each data point conditioned on the weights and parameters. We refer to this type of algorithm as "inter-cluster parallelizable", since the cluster label for each point within a cluster can be sampled in parallel.

The recent works of [83] and [127] present an alternative parallelization scheme for CW samplers in DPMMs and HDPs. They observe that multiple clusters can be grouped into "super-clusters" and that each super-cluster can be sampled independently. We refer to this type of implementation as "intra-cluster parallelizable", since points in different super-clusters can be sampled in parallel, but points within a cluster cannot. This distinction is important as many problems of interest contain far more data points than clusters, and the greatest computational gain may come from inter-cluster parallelizable algorithms. Due to their particular construction, current algorithms group super-clusters solely based on the size of each super-cluster. We will show empirically that this can lead to slow convergence and demonstrate how data-dependent super-clusters improve upon these methods.

There has also been work on collapsed-weight sampling algorithms that consider larger moves to address convergence issues. Green and Richardson [45] present a reversible jump MCMC sampler that proposes splitting and merging components. While a general framework is presented, proposals are model-dependent and generic choices are not specified. Proposed splits are unlikely to fit the posterior since auxiliary variables governing the split cluster parameters and weights are proposed independent of the data. Jain and Neal [63, 64] construct a split by running multiple restricted Gibbs scans for a single cluster in conjugate and non-conjugate models. While each restricted scan improves the constructed split, it also increases the amount of computation needed. As such, it is not easy to determine how many restricted scans are needed. Dahl [27] proposes a split scheme for conjugate models by reassigning labels of a cluster sequentially. All current split samplers construct a proposed move to be used in a Metropolis-Hastings framework. If the split is rejected, considerable computation is wasted, and all information contained in learning the split is forgotten. In contrast, the proposed method of fitting sub-clusters iteratively learns likely split proposals with the auxiliary variables. Additionally, we show that split proposals can be computed in parallel, allowing for very efficient implementations.

Alternatives to classical sampling in DPMMs have been proposed. For example, Liang et al. [79] propose sampling from an augmented space of orderings and consistent partitions. While their algorithm works well when many components exist, they still find that using split-merge algorithms (e.g. [63, 27]) improve convergence speeds.



(a) DPMM Graphical Model

(b) Augmented Super-Cluster

Figure 5.1: Graphical models for the DPMM and augmented super-cluster space. Auxiliary variables are dotted.

■ 5.2 Exact and Parallel Instantiated-Weight Samplers

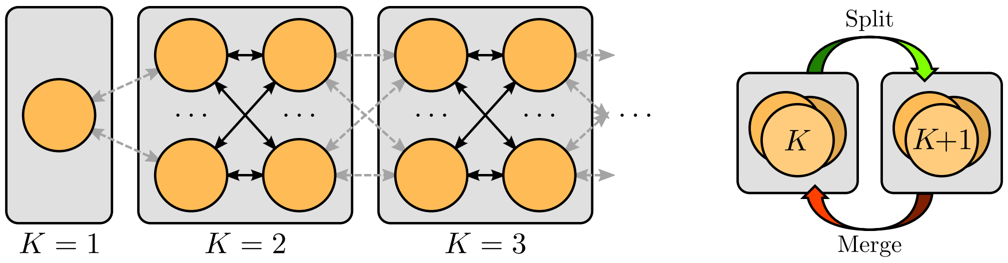
The Dirichlet process mixture model is summarized with previous sampling methods in Section 2.9.2. We reproduce the graphical model corresponding to a DPMM in Figure 5.1a for convenience. As stated previously, sampling from the DPMM is complicated by the infinite length mixture weights, π, and cluster parameters, θ, and often requires using an approximate finite model (such as the finite symmetric Dirichlet or truncated stick-breaking approximations).

We now present an alternative to the instantiated-weight samplers that does not require any finite model approximations. The *detailed balance* property of Definition 2.5.1 underlies most MCMC sampling algorithms. If one desires to sample from a target distribution, satisfying detailed balance for an *ergodic* Markov chain (Definition 2.5.8) guarantees that simulations of the chain will uniquely converge to the target distribution of interest. We now consider the atypical case of simulating from a *non-ergodic* chain with a transition distribution that satisfies detailed balance.

We define a *restricted* sampling algorithm as one that satisfies detailed balance (e.g., using the Metropolis-Hastings or Gibbs sampling algorithms) but that does not result in an ergodic chain. We note that without ergodicity, detailed balance does not imply uniqueness in, or convergence to the stationary distribution. However, multiple restricted samplers can be combined to form an ergodic chain, ensuring the uniqueness of the stationary distribution. In particular, we consider a sampler that is restricted to only sample labels belonging to non-empty clusters. Such a sampler is not ergodic because it cannot create new clusters. However, when mixed with a sampler that proposes splits, the resulting chain is ergodic and yields a valid sampler. A visualization of the state space is shown in Figure 5.2. We now consider a restricted *Gibbs* sampler. The coupled samplers that split or merge clusters is discussed in Sections 5.3-5.4.

■ 5.2.1 Restricted DPMM Gibbs Sampler with Super-Clusters

A property stemming from the definition of Dirichlet processes is that the measure for every finite partitioning of the measurable space is distributed according to a Dirichlet distribution [33]. While the DP places an infinite length prior on the labels, denoted with z, any realization of z will belong to a finite number of clusters. Supposing $z_i \in \{1, \cdots, K\}$, $\forall i$, we showed in Section 2.9.2 that the posterior distribution of mixture



(a) State Diagram of Restricted Sampler (b) State Diagram with Split/Merge

Figure 5.2: Visualizations of the state diagrams for the restricted sampler and the restricted sampler with split/merge moves. Each orange circle represents some configuration of z with K unique values. Dotted arrows correspond to the transitions in the state diagram that are not allowed due to the restricted sampling. This results in isolated islands of states, which the split and merge moves connect.

weights, π, conditioned on the cluster labels follows a Dirichlet distribution, reproduced here as

$$(\pi_1, \cdots, \pi_K, \pi_{K+1}) \sim \text{Dir}\left(\pi_1, \ldots, \pi_K, \tilde{\pi}_{K+1}; N_1, \ldots, N_K, \alpha\right), \tag{5.1}$$

where $N_k = \sum_i \mathbb{1}[z_i = k]$ is the number of points in cluster k, and $\pi_{K+1} = \sum_{k=K+1}^{\infty} \pi_k$ is the sum of all empty mixture weights. This relationship has previously been noted in the literature (c.f. [116]). This leads to the following iterated restricted Gibbs sampler:

$$\begin{aligned} (\pi_1, \ldots, \pi_K, \tilde{\pi}_{K+1}) &\sim \text{Dir}(N_1, \ldots, N_K, \alpha), && && (5.2)\\ \theta_k &\overset{\propto}{\sim} f_x(x_{\mathcal{I}_k}; \theta_k) f_\theta(\theta_k; \lambda), && \forall k \in \{1, \ldots, K\}, && (5.3)\\ z_i &\overset{\propto}{\sim} \sum_{k=1}^{K} \pi_k f_x(x_i; \theta_k) \mathbb{1}[z_i = k], && \forall i \in \{1, \ldots, N\}, && (5.4) \end{aligned}$$

where $\overset{\propto}{\sim}$ denotes drawing a sample from a distribution proportional to the equation on the right and the subscript $\mathcal{I}_k \triangleq \{i; z_i = k\}$ denotes the set of indices with label $z_i = k$. The astute reader may realize that these distributions are quite similar to posterior inference in finite mixture models discussed in Section 2.8 or the finite approximations to the DP discussed in Section 2.9.2. The main difference is that in this formulation, the concentration parameter still corresponds to the probability of creating a new cluster, but z_i is never actually allowed to create a new cluster. These steps preserve detailed balance on the exact model, unlike the finite approximations. Additionally, we note that each of these steps can be parallelized and that this procedure applies to conjugate and non-conjugate priors because the mixture parameters are explicitly represented. When non-conjugate priors are used, any proposal that leaves the stationary distribution invariant can be used (c.f. [93]).

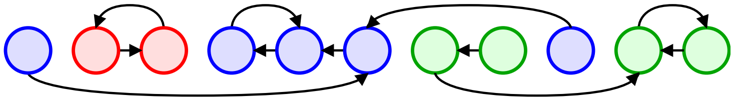


Figure 5.3: An illustration of the super-cluster grouping. Nodes represent clusters, arrows point to neighbors, and colors represent the implied super-clusters.

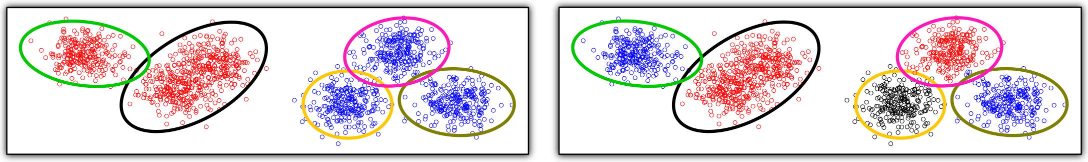


Figure 5.4: An illustration of the difference between data-dependent and data-independent super-clusters. Ellipses indicate cluster means and covariances. Color of data points indicate super-cluster membership. (left) super-clusters from the presented the algorithm. (right) super-clusters from [83].

Data-Dependent Super-Clusters

Similar to previous super-cluster methods, we can also restrict each cluster to only consider moving to a subset of other clusters. The super-clusters of [83] and [127] are formed using a size-biased sampler. This can lead to slower convergence since clusters with similar data may not be in the same super-cluster. Because *any* restricted Gibbs sampler satisfies detailed balance, any algorithm that assigns finite probability to all super-cluster grouping will still satisfy detailed balance.

We therefore can augment the sample space with super-cluster groups, g, that group similar clusters together. The resulting graphical model is depicted in Figure 5.1b. Conditioned on g, Equation 5.4 is altered to only consider labels within the super-cluster that the data point currently belongs to. The super-cluster sampling procedure is described in Algorithm 5.1. Here, D denotes an arbitrary distance measure between probability distributions. In our experiments, we use the symmetric version of KL-divergence (J-divergence). When the J-divergence is difficult to calculate, any distance measure can be substituted. For example, in the case of multinomial distributions, we use the J-divergence for the categorical distribution as a proxy. An illustration of the implied super-cluster grouping from the algorithm is shown in Figure 5.3 and a visualization of an actual super-cluster grouping is shown in Figure 5.4. Notice that the super-cluster groupings using [83] are essentially random while the data-dependent super-clusters from Algorithm 5.1 are grouped by similar data.

■ 5.3 Randomized Split/Merge Moves

The preceding section showed that an exact MCMC sampling algorithm can be constructed by alternating between a restricted Gibbs sampler and split moves. In this

Algorithm 5.1 Sampling Super-clusters with Similar Cluster

```
1. Form the adjacency matrix, $A$, where $A_{k,m} = \exp[-D(f_x(\circ;\theta_k), f_x(\circ;\theta_m))]$
2. For each cluster, $k$, sample a random neighbor $k'$, according to
       $k' \overset{\propto}{\sim} \sum_m A_{k,m} \mathbb{I}[k' = m]$
3. Form the groups of super-clusters, $g$, by finding the separate connected graphs
```

section, we present a pair of data-independent split and merge proposals. Split moves that are constructed without knowledge of the data will typically be nonsensical, and we should not expect these split moves to perform well. Data-independent merge moves, in contrast to splits, can produce sensible proposals because the only way to merge two clusters is to simply put all the data into one new cluster. As we shall see in Section 5.4, these merge moves will have an important role in sampling from the true model.

Similar "randomized" moves were first considered in [63], where a proposed splitting of a cluster was generated by randomly assigning each data point to one of two new clusters with probability 0.5. We will use the symbols $\natural$, $\flat$, and $\sharp$ to denote cluster indices that will be involved in splits and merges, where $\natural, \flat, \sharp \in \{1, \ldots, K\}$. A proposed merge move will merge clusters $\flat$ and $\sharp$ into cluster $\natural$. This notation is motivated by music theory, where a flat accidental ($\flat$) combined with a sharp accidental ($\sharp$) results in a natural note ($\natural$).

We now discuss the randomized splits of [63]. A proposed split of cluster $\natural$ into clusters $\flat$ and $\sharp$, denoted $Q^K_{\text{rsplit-}\natural}$, is first selected with probability $q(Q^K_{\text{rsplit-}\natural})$ and then constructed according to

$$\hat{z}_i \sim q(\hat{z}_i | z, Q^K_{\text{rsplit-}\natural}) = \begin{cases} 1, & \hat{z}_i = z_i, z_i \neq \natural, \\ 0.5, & \hat{z}_i = \flat, z_i = \natural, \\ 0.5, & \hat{z}_i = \sharp, z_i = \natural. \end{cases} \tag{5.5}$$

The corresponding merge move, $Q^{K+1}_{\text{rmerge-}\flat\sharp}$, is selected with probability $q(Q^{K+1}_{\text{rmerge-}\flat\sharp})$ and simply puts all of the data associated with clusters $\flat$ and $\sharp$ into cluster $\natural$. We note that this procedure is slightly different from the original work of [63], but will suffice for our purposes. The Hastings ratio for proposing a merge in this framework is

$$H^{\text{J.N.}}_{\text{merge-}\flat\sharp} = \frac{p(\hat{z})p(x|\hat{z})}{p(z)p(x|z)} \frac{q(z|\hat{z})}{q(\hat{z}|z)} = \frac{\Gamma(N_\flat + N_\sharp)}{\alpha\Gamma(N_\flat)\Gamma(N_\sharp)} \frac{p(x|\hat{z})}{p(x|z)} \frac{0.5^{N_\flat + N_\sharp}}{1} \frac{q(Q^{K-1}_{\text{rsplit-}\natural})}{q(Q^K_{\text{rmerge-}\flat\sharp})} \tag{5.6}$$

This randomized procedure has one minor flaw; corresponding partitions from a split will have high probability of being similarly sized. This is in direct contrast to the Dirichlet process prior on partitions which explicitly favors larger clusters getting larger.

We highlight the discrepancy between the proposal and the prior with an example. Consider the proposal of merging two clusters, m and n, where $N_m = 90$ and $N_n = 10$, with $\alpha = 1$. It would be preferable if the acceptance of the proposed merge was based on the data. However, this proposal has a corresponding Hastings ratio of

$$H^{\text{J.N.}}_{\text{merge-}\flat\sharp} = \frac{p(x|\hat{z})}{p(x|z)} \frac{q(Q^{K-1}_{\text{rsplit-}\natural})}{q(Q^{K}_{\text{rmerge-}\flat\sharp})} e^{-36.635},$$

which extremely favors rejecting the proposal, regardless of the data.

We present a slight improvement over this randomized split proposal that is still data-independent. As stated in Section 2.5.1, the closer the proposal distribution is to the target distribution, the better the convergence. We therefore propose a split by generating from the related Dirichlet-Categorical distribution

$$\hat{z}_{\{\natural\}} \sim \text{DirCat}(\hat{z}_{\{\natural\}}; \tfrac{\alpha}{2}, \tfrac{\alpha}{2}), \tag{5.7}$$

where the subscript $\{\natural\}$ denotes the subset of indices that have $z_i = \natural$. All other values of z_i remain unchanged. Again, the corresponding merge move simply combines clusters $\flat$ and $\sharp$ into one new cluster. This set of split/merge proposals results in the following Hastings ratio for a proposed merge

$$H^{\text{rand}}_{\text{merge-}\flat\sharp} = \frac{\Gamma(N_\flat + N_\sharp)}{\alpha\Gamma(N_\flat)\Gamma(N_\sharp)} \frac{p(x|\hat{z})}{p(x|z)} \frac{\frac{\Gamma(\alpha)}{\Gamma(\alpha+N_\flat+N_\sharp)} \frac{\Gamma(\frac{\alpha}{2}+N_\flat)\Gamma(\frac{\alpha}{2}+N_\sharp)}{\Gamma(\frac{\alpha}{2})\Gamma(\frac{\alpha}{2})}}{1} \frac{q(Q^{K-1}_{\text{rsplit-}\natural})}{q(Q^{K}_{\text{rmerge-}\flat\sharp})}. \tag{5.8}$$

For the example considered above, this Hastings ratio equates to

$$H^{\text{rand}}_{\text{merge-}\flat\sharp} = \frac{p(x|\hat{z})}{p(x|z)} \frac{q(Q^{K-1}_{\text{rsplit-}\natural})}{q(Q^{K}_{\text{rmerge-}\flat\sharp})} e^{-2.363}.$$

This value essentially results in the proposed sampling being accepted if the data likelihood favors the merge and is a consequence of the Dirichlet-Categorical distribution better fitting the Dirichlet process prior.

Furthermore, generating a sample from Equation (5.7) can be parallelized by first sampling an auxiliary Dirichlet random variable, $\tilde{\pi}$ from

$$\tilde{\pi} \sim \text{Dir}(\tilde{\pi}; \tfrac{\alpha}{2}, \tfrac{\alpha}{2}) \tag{5.9}$$

followed by sampling each z_i according to

$$\hat{z}_i \sim \text{Cat}(z_i; \tilde{\pi}), \quad \forall i \in \{i; z_i = \natural\} \tag{5.10}$$

The Hastings ratio for a random merge proposal can additionally be calculated efficiently from summary statistics (e.g., Equation (2.30)) since the two current clusters,

$\flat$ and $\sharp$, are already instantiated. Thus, a merge can be proposed in constant time. The Hastings ratio for a random split proposal depends on the resulting split cluster assignments, $\hat{z}$. Consequently, a random split proposal requires linear time in the size of the cluster. We therefore choose $q(Q^{K}_{\text{rsplit-}\natural}) = 0.01 \times q(Q^{K+1}_{\text{rmerge-}\flat\sharp})$ so that the random split proposals do not take too much computation. We note that any value besides 0.01 could be used without much difference.

We reiterate that the previous restricted Gibbs sampling algorithm can be paired with *any* split/merge framework to produce an exact sampling algorithm for infinite mixture models. While the randomized split proposals described above do not typically fit the data (and are therefore likely to be rejected), the merge proposals work quite well in practice. In the following section, we describe a more sophisticated framework that excels at proposing likely splits.

■ 5.4 Parallel Split/Merge Moves via Sub-Clusters

We now develop efficient data-dependent split moves that are compatible with conjugate and non-conjugate priors and that can be parallelized. The general approach will be to augment the space with auxiliary variables that learn two *sub-clusters* within each cluster of the mixture model. These sub-clusters will contain a likely partition of the data and will be used to propose splits. We note that in any augmented model, samples of the non-auxiliary variables can be obtained by drawing samples from the joint space and simply discarding any auxiliary values.

■ 5.4.1 Augmenting the Space with Auxiliary Variables

Each regular cluster is augmented with two explicit sub-clusters, herein referred to as the "left" and "right" sub-clusters. Each data point is then attributed with a sub-cluster label, $\overline{z}_i \in \{\ell, r\}$, indicating whether it is associated with the left or right sub-cluster. Additionally, each sub-cluster has an associated pair of weights, $\overline{\pi}_k = \{\overline{\pi}_{k\ell}, \overline{\pi}_{kr}\}$, and parameters, $\overline{\theta}_k = \{\overline{\theta}_{k\ell}, \overline{\theta}_{kr}\}$. These auxiliary variables are named in a similar fashion to their regular-cluster counterparts because of the similarities between sub-clusters and regular-clusters. One naïve choice for auxiliary parameter distributions is

$$
\begin{aligned}
p(\overline{\pi}_k) &= \mathrm{Dir}(\overline{\pi}_{k\ell}, \overline{\pi}_{kr}; \tfrac{\alpha}{2}, \tfrac{\alpha}{2}), && (5.11)\\
p(\overline{\theta}_k) &= f_\theta(\overline{\theta}_{k\ell}; \lambda) f_\theta(\overline{\theta}_{kr}; \lambda), && (5.12)\\
p(\overline{z}|\overline{\pi}, \overline{\theta}, x, z) &= \prod_{k=1}^{K} \prod_{i \in \mathcal{I}_k} \sum_{h \in \{\ell, r\}} \frac{\overline{\pi}_{kh} f_x(x_i; \overline{\theta}_{kh})}{Z_i(x, z, \overline{\pi}_k, \overline{\theta}_k)} \mathbb{I}[\overline{z}_i = h], && (5.13)
\end{aligned}
$$

where the normalization term $Z_i(x, z, \overline{\pi}_k, \overline{\theta}_k)$ is defined to be

$$
Z_i(x, z, \overline{\pi}_k, \overline{\theta}_k) \triangleq \overline{\pi}_{k\ell} f_x(x_i; \overline{\theta}_{k\ell}) + \overline{\pi}_{kr} f_x(x_i; \overline{\theta}_{kr}), \tag{5.14}
$$

and is constant with respect to $\overline{z}$. The corresponding graphical model is shown in Figure 5.5a. It would be advantageous if the form of the posterior for the auxiliary variables matched those of the regular-clusters in Equation (5.2)–(5.4). Unfortunately, because the normalization, Z_i, depends on $\overline{\pi}$ and $\overline{\theta}$, this choice of auxiliary distributions results in the following posterior distributions for $\overline{\pi}$ and $\overline{\theta}$

$$p(\overline{\pi}_k|\bullet) \propto \text{Dir}(\overline{\pi}_{k\ell}, \overline{\pi}_{kr}; \tfrac{\alpha}{2}, \tfrac{\alpha}{2}) \prod_{i \in \mathcal{I}_k} Z_i(x, z, \overline{\pi}_k, \overline{\theta}_k)^{-1}, \tag{5.15}$$

$$p(\overline{\theta}_k|\bullet) \propto \prod_{h=\{\ell,r\}} f_\theta(\overline{\theta}_{kh}; \lambda) f_x(x_{\mathcal{I}_{kh}}; \overline{\theta}_{kh}) \prod_{i \in \mathcal{I}_k} Z_i(x, z, \overline{\pi}_k, \overline{\theta}_k)^{-1}, \tag{5.16}$$

where conditioning on $\bullet$ denotes conditioning on all other variables, and $\mathcal{I}_{kh} \triangleq \{i; z_i = k, \overline{z}_i = h\}$. These posterior distributions are quite different from the regular-cluster posterior distributions, and it is not clear how to sample from these efficiently. We note that this problem only arises in the auxiliary space where x generates the auxiliary label $\overline{z}$ (in contrast to the regular space, where z generates x).

Consequently, we alter the distribution over sub-cluster parameters to be

$$p(\overline{\theta}_k|x, z, \overline{\pi}) \propto f_\theta(\overline{\theta}_{k\ell}; \lambda) f_\theta(\overline{\theta}_{kr}; \lambda). \prod_{i \in \mathcal{I}_k} Z_i(x, z, \overline{\pi}_k, \overline{\theta}_k). \tag{5.17}$$

It is easily verified that this results in the following conditional posterior distributions

$$p(\overline{\pi}_k|\bullet) = \text{Dir}(N_{k\ell} + \alpha/2, N_{kr} + \alpha/2), \qquad \forall k \in \{1, \dots, K\}, \tag{5.18}$$

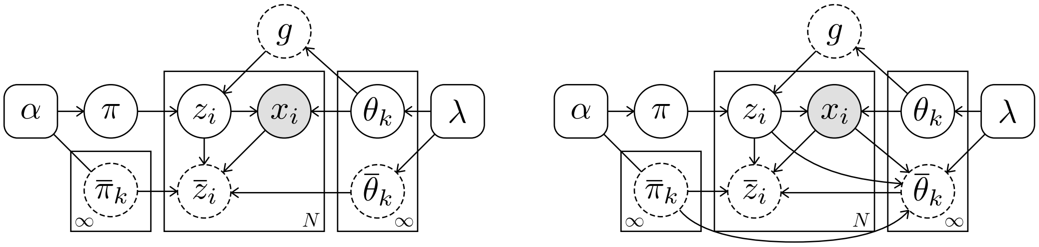
$$p(\overline{\theta}_{kh}|\bullet) \propto f_x(x_{\mathcal{I}_{kh}}; \overline{\theta}_{kh}) f_\theta(\overline{\theta}_{kh}; \lambda), \qquad \forall k \in \{1, \dots, K\}, \forall h \in \{\ell, r\}, \tag{5.19}$$

$$p(\overline{z}_i|\bullet) \propto \textstyle\sum_{h \in \{\ell, r\}} \overline{\pi}_{z_i h} f_x(x_i; \overline{\theta}_{z_i h}) \mathbb{I}[\overline{z}_i = h], \qquad \forall i \in \{1, \dots, N\}, \tag{5.20}$$

which essentially match the distributions for regular-cluster parameters in Equation (5.2)–(5.4). We note that the joint distribution over the augmented space cannot be expressed analytically as a result of only specifying Equation (5.17) up to a proportionality constant that depends on $\overline{\pi}$, x, and z. The corresponding graphical model is shown in Figure 5.5b. Additional details and derivations for this section can be found in Appendix B.

■ 5.4.2 Restricted Gibbs Sampling in Augmented Space

Restricted sampling in the augmented space can be performed in a similar fashion as before. One can draw a sample from the space of K regular clusters by sampling all the regular- and sub-cluster parameters conditioned on labels and data from Equations (5.2), (5.3), (5.18), and (5.19). Conditioned on these parameters, one can sample a regular-cluster label followed by a sub-cluster label for each data point from Equations (5.4) and (5.20). All of these steps can be computed in parallel. The procedure is summarized in Algorithm 5.2.



(a) Naïve Augmented Sub-Cluster Model (b) Correct Augmented Sub-Cluster Model

Figure 5.5: Graphical models for the augmented DPMMs. Auxiliary variables are dotted.

Algorithm 5.2 Restricted Sampling with Sub-Clusters

1. Sample π and $\overline{\pi}$ from Equations (5.2) and (5.18).
2. For each cluster k, sample θ_k and $\overline{\theta}_k$ from Equations (5.3) and (5.19).
3. For each index i, sample z_i and $\overline{z}_i$ from Equation (5.4) and (5.20).

The resulting inference from the restricted Gibbs sampling algorithm for a synthetic Gaussian mixture model is shown in Figure 5.6. We have initialized the inference to have 4 clusters. Inferred regular-cluster parameters are illustrated with a solid ellipse and inferred sub-cluster parameters are illustrated with dotted ellipses of the same color. Because split moves have not been incorporated into the procedure yet, the result is not a valid sample from the posterior. This is indicated by the black and yellow clusters, which each contain two true clusters. However, the dotted ellipses show that the inferred sub-clusters correctly capture the information of interest in representing a likely split.

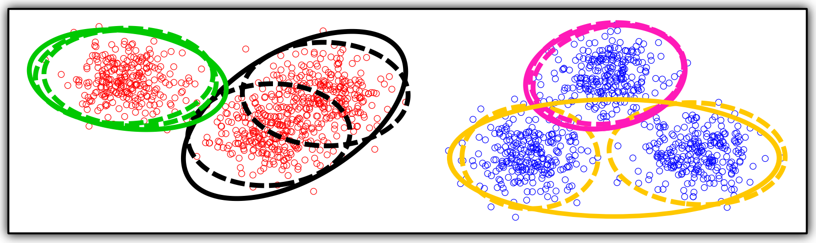


Figure 5.6: A visualization of the inferred sub- and super-clusters of the algorithm. Solid ellipses indicate regular-cluster means and covariances and dotted ellipsses indicate sub-cluster means and covariances. Color of data points indicate super-cluster membership.

■ 5.4.3 Sub-Cluster Split Moves

We now exploit these auxiliary variables to propose likely splits. Similar to previous split/merge algorithms, we use a Metropolis-Hastings (MH) MCMC [51] method for proposed splits. A new set of random variables, $\{\hat{\pi}, \hat{\theta}, \hat{z}, \hat{\overline{\pi}}, \hat{\overline{\theta}}, \hat{\overline{z}}\}$ are proposed via some proposal distribution, q, and accepted with probability

$$\min[1, H] = \min\left[1, \frac{p(\hat{\pi}, \hat{z}, \hat{\theta}, x)p(\hat{\overline{\pi}}, \hat{\overline{\theta}}, \hat{\overline{z}}|x, \hat{z})}{p(\pi, z, \theta, x)p(\overline{\pi}, \overline{\theta}, \overline{z}|x, z)} \cdot \frac{q(\pi, z, \theta, \overline{\pi}, \overline{\theta}, \overline{z}|\hat{\pi}, \hat{z}, \hat{\theta}, \hat{\overline{\pi}}, \hat{\overline{\theta}}, \hat{\overline{z}})}{q(\hat{\pi}, \hat{z}, \hat{\theta}, \hat{\overline{\pi}}, \hat{\overline{\theta}}, \hat{\overline{z}}|\pi, z, \theta, \overline{\pi}, \overline{\theta}, \overline{z})}\right]. \quad (5.21)$$

Unfortunately, because the joint likelihood for the augmented space cannot be expressed analytically, the Hastings ratio for an arbitrary proposal distribution cannot be computed. We now discuss a very specific proposal distribution which results in a tractable Hastings ratio. A split or merge move, denoted by Q, is first selected at random. As we detail shortly, all possible splits and a large subset of all possible merges are considered at every iteration. A randomized proposal can be used instead when the number of clusters is large.

Conditioned on $Q = Q^K_{\text{split-}\natural}$, which splits cluster $\natural$ into clusters $\flat$ and $\sharp$, or $Q = Q^K_{\text{merge-}\flat\sharp}$, which merges clusters $\flat$ and $\sharp$ into cluster $\natural$, a new set of variables are sampled with the following

$$\begin{aligned}
&Q = Q^K_{\text{split-}\natural} && Q = Q^K_{\text{merge-}\flat\sharp} & \\
&\hat{z} = \text{split-}\natural(z, \overline{z}), && \hat{z} = \text{merge-}\flat\sharp(z), & (5.22)\\
&(\hat{\pi}_\flat, \hat{\pi}_\sharp) = \pi_\natural \cdot (u_\flat, u_\sharp), \quad (u_\flat, u_\sharp) \sim \text{Dir}(\hat{N}_\flat, \hat{N}_\sharp), && \hat{\pi}_\natural = \hat{\pi}_\flat + \hat{\pi}_\sharp, & (5.23)\\
&(\hat{\theta}_\flat, \hat{\theta}_\sharp) \sim q(\hat{\theta}_\flat, \hat{\theta}_\sharp|x, \hat{z}, \hat{\overline{z}}), && \hat{\theta}_\natural \sim q(\hat{\theta}_\natural|x, \hat{z}, \hat{\overline{z}}), & (5.24)\\
&\hat{\overline{v}}_\flat, \hat{\overline{v}}_\sharp \sim p(\hat{\overline{v}}_\flat, \hat{\overline{v}}_\sharp|x, \hat{z}), && \hat{\overline{v}}_\natural \sim p(\hat{\overline{v}}_\natural|x, \hat{z}). & (5.25)
\end{aligned}$$

Here, $\overline{v}_k = \{\overline{\pi}_k, \overline{\theta}_k, \overline{z}_{\mathcal{I}_k}\}$ denotes the set of auxiliary variables for cluster k, the function split-$\natural(\circ)$ splits the labels of cluster $\natural$ *deterministically* based on the sub-cluster labels according to

$$\hat{z}_i = \text{split-}\natural(z_i, \overline{z}_i) = \begin{cases} z_i, & z_i \neq \natural \\ \flat, & z_i = \natural, \overline{z}_i = \ell \\ \sharp, & z_i = \natural, \overline{z}_i = r \end{cases}. \quad (5.26)$$

and merge-$\flat\sharp(\circ)$ merges the labels of clusters $\flat$ and $\sharp$ according to

$$\hat{z}_i = \text{merge-}\flat\sharp(z_i) = \begin{cases} z_i, & z_i \neq \flat, z_i \neq \sharp \\ \natural & z_i = \flat \text{ or } z_i = \sharp \end{cases}. \quad (5.27)$$

The proposal of cluster parameters in Equation (5.24) is written in a general form for compatibility with non-conjugate priors. If conjugate priors are used, Equation (5.24) should be sampled directly from the posterior distribution. Sampling auxiliary variables from Equation (5.25) will be discussed shortly. Assuming that this can be performed, we show in Appendix B that the resulting Hastings ratio for a split is

$$\begin{aligned} H^{\text{det}}_{\text{split-}\natural} &= \frac{q(Q^{K+1}_{\text{merge-}\flat\sharp})}{q(Q^{K}_{\text{split-}\natural})} \frac{\alpha q(\theta_\natural | x, z, \hat{z})}{\Gamma(N_\natural) f_\theta(\theta_\natural; \lambda) f_x(x_{\mathcal{I}_\natural}; \theta_\natural)} \prod_{k \in \{\flat, \sharp\}} \frac{\Gamma(\hat{N}_k) f_\theta(\hat{\theta}_k; \lambda) f_x(x_{\mathcal{I}_k}; \hat{\theta}_k)}{q(\hat{\theta}_k | x, z, \hat{z})} \\ &= \frac{q(Q^{K+1}_{\text{merge-}\flat\sharp})}{q(Q^{K}_{\text{split-}\natural})} \frac{\alpha}{\Gamma(N_\natural) f_x(x_{\mathcal{I}_\natural}; \lambda)} \prod_{k \in \{\flat, \sharp\}} \Gamma(\hat{N}_k) f_x(x_{\mathcal{I}_k}; \lambda). \end{aligned} \tag{5.28}$$

The first expression can be used for non-conjugate models, and the second expression can be used in conjugate models where new cluster parameters are sampled directly from the posterior distribution. We note that these expressions do not have any residual normalization terms and can be computed exactly, even though the joint distribution of the augmented space can not be expressed analytically.

As noted previously, we consider every possible split every each iteration, resulting in $q(Q^K_{\text{split-}\natural}) = 1$. When proposing merge moves, we construct $\lfloor K/2 \rfloor$ possible pairs by first generating a random permutation of the integers in $[1, K]$, and proposing to merge disjoint neighbors. For example, if the random permutation for $K = 7$ is $\{\overbrace{3\,1}\ \overbrace{7\,4}\ \overbrace{2\,6}\ 5\}$, we will propose to merge topics 3 and 1, topics 7 and 4, and topics 2 and 6. It is easily verified that the probability of proposing any specific merge is then $\frac{2\lfloor K/2 \rfloor}{K(K-1)}$. The probability of selecting any specific merge is approximately $\frac{2}{K}$, meaning that it is uncommon to select a pair of clusters that should actually be merged. We therefore use this proposal K times, resulting in $q(Q^K_{\text{merge-}\flat\sharp}) = \frac{2\lfloor K/2 \rfloor}{K-1}$.

Unfortunately, the Hastings ratio for a merge move is slightly more complicated. We discuss these complications following the explanation of sampling the auxiliary variables in the next section.

■ 5.4.4 Deferred Metropolis-Hastings Sampling

The preceding section showed that sampling a split according to Equations (5.22)–(5.25) results in an accurate MH framework. However, sampling the auxiliary variables from Equation (5.25) is not straightforward. This step is equivalent to sampling cluster parameters and labels for a 2-component mixture model, which is known to be difficult. One typically samples from this space using an MCMC procedure. In fact, that is precisely what the restricted Gibbs sampler is doing. We therefore sample from Equation (5.25) by running a restricted Gibbs sampler for each newly proposed sub-cluster until they have burned-in. We monitor the data-likelihood for cluster k, $\overline{\mathcal{L}}_k = f_x(x_{\mathcal{I}_{k\ell}}; \overline{\theta}_{k,\ell}) \cdot f_x(x_{\mathcal{I}_{kr}}; \overline{\theta}_{k,r})$ and declare burn-in once $\overline{\mathcal{L}}_k$ begins to oscillate.

Furthermore, due to the implicit marginalization of auxiliary variables, the restricted Gibbs sampler and split moves that act on clusters that were not recently split do not depend on the proposed auxiliary variables. As such, these proposals can be computed before the auxiliary variables are even proposed. The sampling of auxiliary variables of a recently split cluster are *deferred* to the restricted Gibbs sampler while the other sampling steps are run concurrently. Once a set of proposed sub-clusters have burned-in, the corresponding clusters can be proposed to split again.

■ 5.4.5 Merge Moves with Random Splits

The Hastings ratio for a merge depends on the proposed auxiliary variables for the reverse split. Since proposed splits are deterministic conditioned on the sub-cluster labels, the Hastings ratio will be zero if the proposed sub-cluster labels for a merge do not match those of the current clusters. We show in Appendix B.3 that as the number of data points grows, the acceptance ratio for a merge move quickly decays. With only 256 data points, the acceptance ratio for a merge proposal for 1000 trials in a 1D Gaussian mixture model did not exceed 10^{-16}. We therefore approximate all merges with an automatic rejection. Unfortunately, this can lead to slow convergence when too many clusters exist at the current iteration.

The sub-cluster split and merge moves excel and proposing good split moves but are poor at accepting merge moves. We remind the reader that the pair of randomized split and merge proposals presented in Section 5.3 is essentially the opposite; randomized splits are typically poor but the corresponding randomized merges often perform well. Therefore, we mix the sub-cluster split sampler with the randomized split/merge sampler to achieve good splits *and* merges.

■ 5.5 Non-Deterministic Sub-Cluster Split Proposals

The preceding section presented a sampling algorithm that samples from the exact model without any approximations. While we have empirically noticed that this sampling algorithm works very well in practice, it is a bit unnerving that the sub-cluster merges are always rejected and that the randomized splits and merges are needed to achieve good convergence. In this section, we propose an alternative set of split/merge moves, where both the split and the merge are likely to be accepted. We note that this algorithm will require a slight approximation whereas the previous framework did not.

Instead of deterministically copying the sub-topic labels, we modify the proposal to *sample* a split. The sub-cluster statistics are used to propose a new cluster assignment by first constructing temporary parameters, $\{\tilde{\pi}_\flat, \tilde{\pi}_\sharp, \tilde{\theta}_\flat, \tilde{\theta}_\sharp\}$

$$(\tilde{\pi}_\flat, \tilde{\pi}_\sharp) = \pi_\natural \cdot (\overline{\pi}_{\natural\ell}, \overline{\pi}_{\natural r}), \quad (\tilde{\theta}_\flat, \tilde{\theta}_\sharp) = (\overline{\theta}_{\natural\ell}, \overline{\theta}_{\natural r}). \tag{5.29}$$

Conditioned on these temporary cluster parameters, new cluster assignments for topic $\natural$ are drawn from

$$q(\hat{z}|v, \overline{v}, Q^K_{\text{split-}\natural}) = \prod_{i \in \mathcal{I}_\natural} \sum_{k \in \{\flat, \sharp\}} \frac{\tilde{\pi}_k f_x(x_i; \tilde{\theta}_k) \mathbb{I}[\hat{z}_i = k]}{\tilde{\pi}_\flat f_x(x_i; \tilde{\theta}_\flat) + \tilde{\pi}_\sharp f_x(x_i; \tilde{\theta}_\sharp)}. \tag{5.30}$$

We note that a sample from this distribution is already drawn from the restricted Gibbs sampler described in Equation (5.20). Therefore, no additional computation is needed to sample from this distribution. If the split is rejected, the $\hat{z}$ is used as the next sample of the auxiliary $\overline{z}$ for cluster $\natural$.

The corresponding merge move combines topics $\flat$ and $\sharp$ into topic $\natural$ by deterministically performing

$$q(\hat{z}_i|v, Q^K_{\text{merge-}\flat\sharp}) = \mathbb{I}[\hat{z}_i = \natural], \quad \forall i \in \mathcal{I}_\flat \cup \mathcal{I}_\sharp. \tag{5.31}$$

Split and merge proposals for $\hat{\pi}$, $\hat{\theta}$, and $\hat{\overline{v}}$ follow the previous distributions of Equations (5.23)–(5.25) conditioned on $\hat{z}$.

The resulting Hastings ratio for this non-deterministic split only differs from Equation (5.28) by including the additional term from Equation (5.30) and can be expressed as

$$H^{\text{non-det}}_{\text{split-}\natural} = H^{\text{det}}_{\text{split-}\natural} \frac{1}{q(\hat{z}|v, \overline{v}, Q^K_{\text{split-}\natural})} \tag{5.32}$$

We record the probability $q(\hat{z}|v, \overline{v})$ when generating the auxiliary variables, and all remaining terms can be computed efficiently as before without iterating through the data.

The Hastings ratio for a merge is essentially the reciprocal of Equation 5.32. However, calculating the terms for a merge is slightly more problematic since the probability of the reverse split after a merge is proposed, $q(z|\hat{v}, \hat{\overline{v}}, Q^{K-1}_{\text{split-}\natural})$, depends on the inferred sub-cluster parameters, $\hat{\overline{v}} = \{\hat{\overline{\pi}}, \hat{\overline{\theta}}\}$. These proposed sub-topic parameters are not readily available due to the Deferred Metropolis-Hastings. Instead, we calculate the Hastings ratio by approximating the inferred sub-clusters with the two original clusters that are merging. In the limit, as the Markov chain has reached its stationary distribution, this assumption is quite accurate because of the similarity between regular-clusters and sub-clusters.

With this approximation, generating the reverse move that splits cluster l into m and n can be expressed as

$$q(z|\hat{v}, \hat{\overline{v}}, Q^{K-1}_{\text{split-}\natural}) \approx \prod_{i \in \mathcal{I}_\flat \cup \mathcal{I}_\sharp} \frac{\pi_{z_i} f_x(x_i; \theta_{z_i})}{\sum_{k \in \{\flat, \sharp\}} \pi_k f_x(x_i; \theta_k)}. \tag{5.33}$$

All the terms in this ratio are already calculated in the restricted Gibbs steps of Equation (5.4) in Algorithm 5.2. When aggregated properly, any merge can be proposed in constant time. We maintain a $K \times K$ matrix $\mathcal{L}$, where each element aggregates the following

$$\mathcal{L}_{kk} = \prod_{i \in \mathcal{I}_k} \pi_k f_x(x_i; \theta_k), \tag{5.34}$$

$$\mathcal{L}_{kl} = \prod_{i \in \mathcal{I}_k} \sum_{\kappa \in \{k,l\}} \pi_\kappa f_x(x_i; \theta_\kappa). \tag{5.35}$$

The reverse split move can then be approximated with

$$q(z|\hat{v}, \hat{\bar{v}}, Q^{K-1}_{\text{split-}\natural}) \approx \frac{\mathcal{L}_{\flat\flat}\mathcal{L}_{\sharp\sharp}}{\mathcal{L}_{\flat\sharp}\mathcal{L}_{\sharp\flat}}. \tag{5.36}$$

This concludes the discussion of non-deterministic split and merge proposals in DPMMs. Because merge proposals are now accepted with non-zero probability, incorporating randomized split/merge proposals is no longer needed.

5.6 Experimental Results

In this section, we analyze the proposed sampling method. We compare the different split/merge proposals described in the preceding sections and compare the proposed methods to other popular MCMC sampling methods.

5.6.1 Split/Merge Proposal Comparison

We begin by comparing the different split/merge proposals described in the preceding sections. We consider four algorithms: using deterministic and randomized proposals (Det+Rand), using only deterministic proposals (Det), using only randomized proposals (Rand), and using the non-deterministic proposals (Non-Det). We test the methods on Gaussian model with a Normal Inverse-Wishart prior on the MNIST dataset [76] by first running PCA on the 70,000 training and test images to 50 dimensions. We additionally test the algorithm on multinomial data with a Dirichlet prior on the Associated Press [9] (2,246 documents and 10,473 dimension dictionary). Results are shown in Figure 5.7 for each dataset with 1, 50, and 100 initial clusters. Each plot shows the average log likelihood for multiple sample paths obtained using 16 cores.

The plots show a few important points. The Rand algorithm does not typically propose useful splits, causing the log-likelihood to stay constant when initialized to a single cluster. Conversely, the Det algorithm does not accept merges, causing the log-likelihood to prematurely plateau when initialized to too many clusters. The Det+Rand and Non-Det methods both do not suffer from these issues since likely splits are proposed, and merges are accepted with non-negligible probability. Additionally, the Non-Det method seems to take slightly longer than Det+Rand. This increase in time is due to the extra computation required to aggregate statistics in $\mathcal{L}$ for the non-deterministic moves. For this reason, the remainder of this chapter will use the Det+Rand split proposals. However, as we shall see in Chapter 6, the non-deterministic proposals will still play an important role for extensions to hierarchical models.

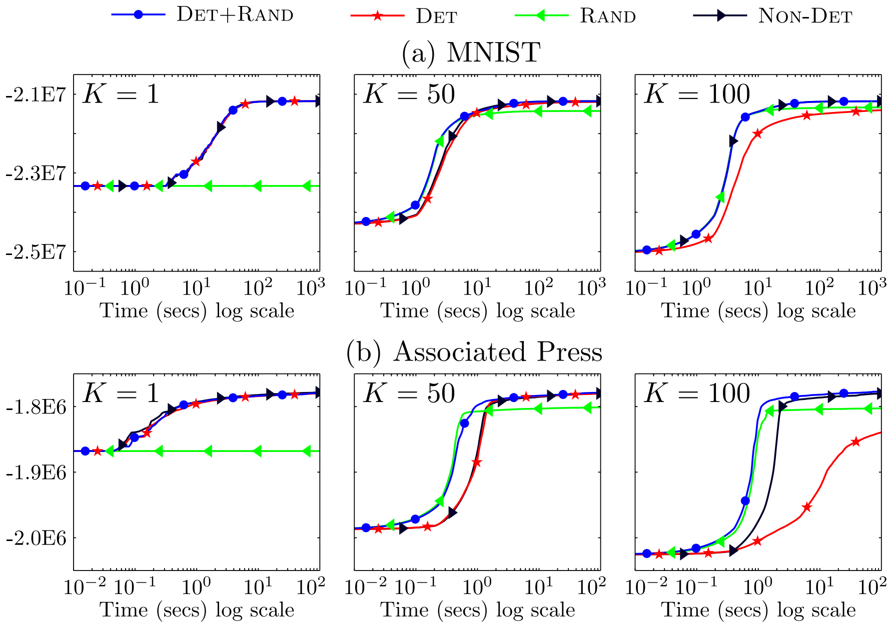


Figure 5.7: Log likelihood vs. computation time for various split/merge proposals on two datasets. Three different initializations with varying number of initial clusters are used for each algorithm.

■ 5.6.2 Parallelizability and Sensitivity to Hyper-Parameters

Next, we compare the proposed algorithm with other MCMC sampling algorithms on synthetic data. We consider three different versions of the proposed algorithm: using sub-clusters with and without super-clusters (SUBC and SUBC+SUPC) and an approximate method that does not wait for the convergence of sub-clusters to split (SUBC+SUPC APPROX). We note that while we do not expect this last version to converge to the correct distribution, empirical results show that it is similar in average performance. We compare the proposed methods against four other methods: the finite symmetric Dirichlet approximate model (FSD) with 100 components, a Rao-Blackwellized Gibbs sampler (GIBBS), a Rao-Blackwellized version of the original super-cluster work of [83] (GIBBS+SUPC), and the current state-of-the-art split/merge sampler [27] (GIBBS+SAMS). In our implementations, the concentration parameter is not resampled, though one could easily use a slice-sampling algorithm if desired.

We compare these algorithms on synthetic Gaussian data with a Normal Inverse-Wishart prior. 100,000 data points are simulated from ten 2D Gaussian clusters. The average log likelihood for multiple sample paths obtained using the algorithms without parallelization for different numbers of initial clusters K and concentration parameters α are shown in the first two columns of Figure 5.8. In this high data regime, α should have little effect on the resulting clusters. However, we find that the samplers without split/merge proposals (FSD, GIBBS, GIBBS+SC) perform very poorly when the initial

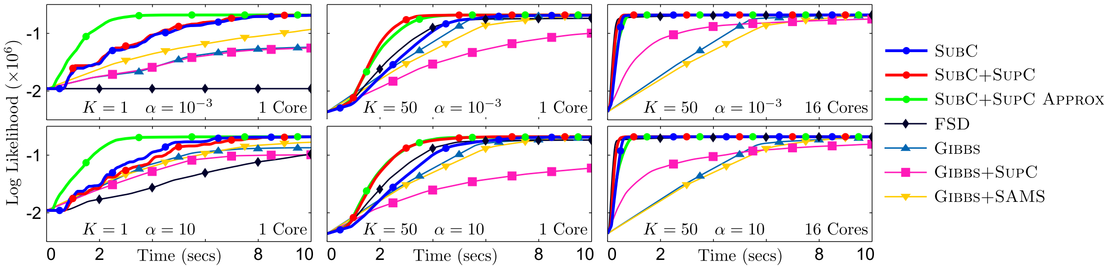


Figure 5.8: Results on synthetic data for various initial clusters K, concentration parameters α, and cores.

number of clusters and the concentration parameter is small. We also find that the super-cluster method, Gibbs+SC, performs even worse than regular Gibbs sampling. This is likely due to the super-clusters not being grouped by similar data, which hinders convergence because data points cannot move between different super-clusters. In contrast, the proposed super-cluster method does not suffer from the same convergence problems, but is comparable to SubC because there are a small number of clusters. Finally, the approximate sub-cluster method has significant gains when only one initial cluster is used, but performs approximately the same with more initial clusters.

Next we consider parallelizing the algorithms using 16 cores in the last column of Figure 5.8. The four inter-cluster parallelizable algorithms, SubC, SubC+SupC, SubC+SupC Approx, and FSD exhibit an order of magnitude speedup, while the the intra-cluster parallelizable algorithm Gibbs+SupC only has minor gains. As expected, parallelization does not aid the convergence of algorithms, only the speed at which they converge.

■ 5.6.3 Real-World Datasets

We now show results on real data. We test a Gaussian model with a Normal Inverse-Wishart prior on the MNIST dataset [76] by first running PCA on the 70,000 training and test images to 50 dimensions. Results on the MNIST dataset are shown in Figure 5.9a. We additionally test the algorithm on multinomial data with a Dirichlet prior on the following datasets: Associated Press [9] (2,246 documents and 10,473 dimension dictionary), Enron Emails [2] (39,861 documents and 28,102 dimension dictionary), New York Times articles [2] (300,000 documents and 102,660 dimension dictionary), and PubMed abstracts [2] (8,200,000 documents and 141,043 dimension dictionary). Results are shown in Figure 5.9b-e. In contrast to HDP models, each document is treated as a *single* draw from a multinomial distribution. We note that on the PubMed dataset, we had to increase the approximation of FSD to 500 components after observing that SubC inferred approximately 400 clusters. On real data, it is clearly evident that the other algorithms have issues with convergence. In fact, in the allotted time, no algorithms besides the proposed methods converge to the same log likelihood with the

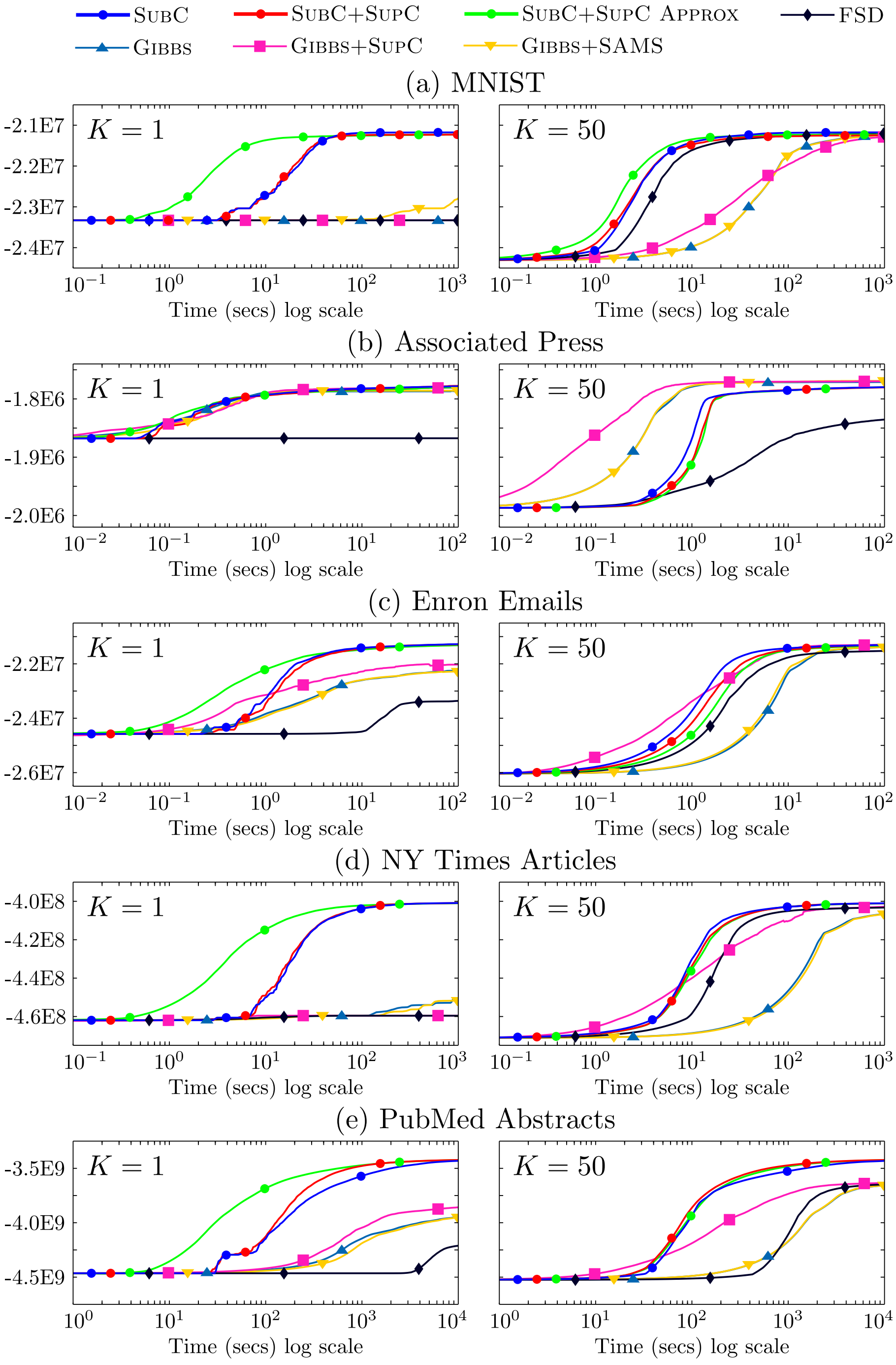


Figure 5.9: Results on real-world Gaussian and Multinomial data. Each figure plots log likelihood vs. computation time. All parallel algorithms use 16 cores. The first column is initialized to one cluster and the second column is initialized to 50 clusters.

two different initializations on the larger datasets. The presented sub-cluster methods converge faster to a better sample than other algorithms converge to a worse sample.

On the small, Associated Press dataset, the proposed methods actually perform slightly worse than the GIBBS methods. Approximately 20 clusters are inferred for this dataset, resulting in approximately 100 observations for each cluster. In these small data regimes, it is important to marginalize over as many variables as possible. We believe that because the GIBBS methods marginalize over the cluster parameters and weights, they achieve better performance as compared to the sub-cluster methods and FSD which explicitly instantiate them. This is not an issue with larger datasets.

■ 5.7 Discussion

This chapter develops a new MCMC sampling algorithm for Dirichlet process mixture models that we call the DP Sub-Cluster algorithm. The DP Sub-Cluster algorithm is easily parallelized to scale to large datasets, and exhibits better convergence through the proposed sub-cluster split and merge moves. In fact, the algorithm converges to a better configuration than other algorithms, and does so approximately 10–10^3 times faster. While the focus of discussion in this chapter was on DPMMs, we believe that the framework used in developing the DP Sub-Cluster algorithm is widely applicable to the general finite or infinite mixture model.

The deterministic split move paired with randomized split and merge proposals is paired with a restricted Gibbs sampling algorithm to produce an MCMC algorithm that does not require any approximations. However, the non-deterministic split and merge moves required a slight approximation due to the deferred Metropolis-Hastings. In particular, because the sub-cluster assignments are not readily available after a proposed split or merge, the sub-clusters were estimated using the original clusters. While this approximation was argued to be fairly accurate in the limit of reaching the stationary distribution, the theory would benefit from a more rigorous analysis.

Additionally, we note that small clusters, which often appear from the collapsed Gibbs sampling algorithms, tend to occur less frequently in the DP Sub-Cluster algorithm. This is likely due to the implicit bias towards proposing large split moves from the instantiated sub-clusters paired with the restricted Gibbs sampling algorithm that does not allow the creation of new clusters. This is in stark contrast to the collapsed Gibbs sampling algorithms, which can start a cluster with a single data point at any iteration. As shown in [91], these small clusters are in the typical set of DPMMs, and should be expected.

We also believe method of deferred Metropolis-Hastings, where the auxiliary variables are deferred tot he restricted Gibbs sampling algorithm, is an interesting idea that other auxiliary variable methods may benefit. This type of proposal could also benefit from a more detailed derivation.

While we have done our best to compare many of the current MCMC sampling algorithms, one additional interesting experiment that could be performed is comparing the

DP Sub-Cluster algorithm to variational inference methods. In particular, variational methods often benefit from being highly parallelizable. Moreover, recent variational methods (e.g., [57]) also exploit split/merge moves, but benefit because they do not need to satisfy detailed balance.

We believe one promising direction is the extension of the DP Sub-Cluster algorithm to models that use Dirichlet processes as a component in a larger model. For example, the Hierarchical Dirichlet Process [116], the Hierarchical Dirichlet Process Hidden Markov Model [34, 116], and the Dependent Dirichlet Process [81] could all greatly benefit from scalable MCMC inference methods. In the next chapter, we demonstrate how the DP Sub-Cluster algorithm can be extended to the HDP, with the intent to show many models can take advantage of the work presented in this chapter..