Scalable and Flexible Clustering of Grouped Data via Parallel and Distributed Sampling in Versatile Hierarchical Dirichlet Processes

Or Dinari and **Oren Freifeld**
The Department of Computer Science, Ben-Gurion University
dinari@post.bgu.ac.il, orenfr@cs.bgu.ac.il

Abstract

Adaptive clustering of grouped data is often done via the Hierarchical Dirichlet Process Mixture Model (HDPMM). That approach, however, is limited in its flexibility and usually does not scale well. As a remedy, we propose another, but closely related, hierarchical Bayesian nonparametric framework. Our main contributions are as follows. 1) a new model, called the Versatile HDPMM (vHDPMM), with two possible settings: full and reduced. While the latter is akin to the HDPMM's setting, the former supports not only global features (as HDPMM does) but also local ones. 2) An effective mechanism for detecting global features. 3) A new sampler that addresses the challenges posed by the vHDPMM and, in the reduced setting, scales better than HDPMM samplers. 4) An efficient, distributed, and easily-modifiable implementation that offers more flexibility (even in the reduced setting) than publicly-available HDPMM implementations. Finally, we show the utility of the approach in applications such as image cosegmentation, visual topic modeling, and clustering with missing data.

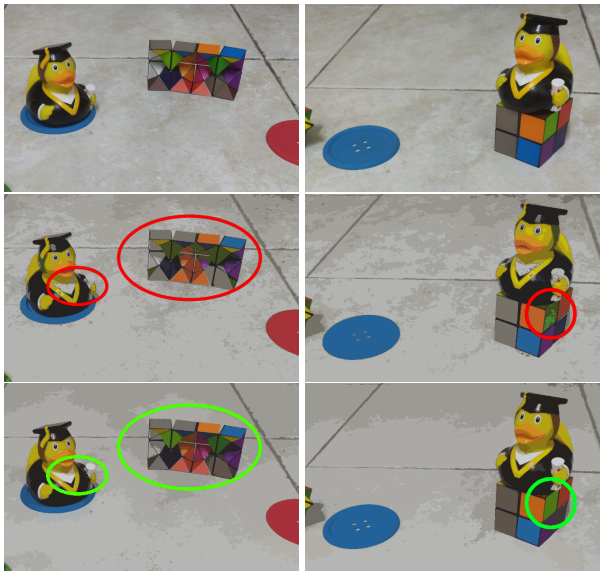

Figure 1: An image cosegmentation example. Top row: The original images (2 out of 5 are shown). Middle: HDPMM cosegmentation (based on global color features). Bottom: vHDPMM cosegmentation (based on both global color features and local spatial features). Compared with the vHDPMM, the HDPMM fails in separating infrequent colors (*e.g.*, the whiteness of the duck's diploma and shirt is mistaken to be grey) and also yields too-grainy results. The full images and results are in the **Sup. Mat.**

1 INTRODUCTION

The Hierarchical Dirichlet Process Mixture Model (HDPMM) [39] clusters grouped data when the number of clusters is unknown. In this paper we note, and then solve, three problems associated with the HDPMM. First, while the HDPMM allows for group-specific mixture weights, it cannot capture various types of other realistic inter-group differences. Second, traditional HDPMM inference methods scale poorly w.r.t. the number of data points per group. Third, most available HDPMM implementations are not distributed and/or limited to (a specific form of) topic modeling. With these considerations in mind, this paper proposes a new model along with a new sampler whose distributed and flexible implementation we make publicly available. Our model has two settings; while its

Acknowledgements. This work was partially funded by the Lynn and William Frankel Center for Computer Science at BGU. Or Dinari was also funded in part by the Jabotinsky Scholarship from Israel's Ministry of Technology and Science, and by BGU's Hi-Tech Scholarship.

Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence (UAI), PMLR volume 124, 2020.

reduced setting is similar to that of the HDPMM, its *full setting* is more general, as we now describe. There are J data groups, where, for $j \in (1, \ldots, J)$, group j consists of n_j data points and is denoted by $\boldsymbol{r}_j = (r_{ji})_{i=1}^{n_j}$. The goal is to cluster all the data (across all groups) into K clusters (K being unknown), under the assumption that each data point r_{ji}, also known as a *feature vector*, has two parts, $r_{ji} = (x_{ji}, y_{ji})$, where x_{ji} consists of features that are expected to help clustering when all the groups together are treated as one big group, and y_{ji} consists of features that are more useful when each $\boldsymbol{r}_j$ is considered separately. We refer to x_{ji} and y_{ji} as *global* and *local* features, respectively. The reduced setting (*i.e.*, that of HDPMM), is obtained when there are no local features (*i.e.*, $r_{ji} = x_{ji}$ for every j and every i). To fix ideas, consider the following computer-vision *cosegmentation* task (Fig. 1): given J images, partition each image into segments *consistently across the images*; *i.e.*, if the same (possibly-disconnected) segment appears in more than one image, the results should reflect this. One difficulty is that the locations, as well as the number of the segments, are unknown and vary with each image. Let the chosen approach be a spatio-color clustering via a mixture model (for related non-hierarchical spatio-color mixture models in computer vision, see, *e.g.*, [8, 19, 1, 9, 18, 42]). Here, each image is a data group where each pixel is associated with a 5D data point = (a 3D RGB value, a 2D location). The color is the global part as the colors of an object are often similar across different images, while the location is the local part as an object's location usually varies across images. Thus, while it is sensible to model colors using a global mixture model (*i.e.*, across all images, perhaps while allowing different mixture weights in each image), a local model (*i.e.*, image-specific) is more suitable to differentiate between objects based on their locations. This intuitive observation will be proven empirically in § 4.

Back to our general setting, there is usually a *tying* between the local and global parts (*e.g.*, in the cosegmentation example, the locations of parts of a certain object in different images are tied to it, and thus are implicitly also tied to its color(s)). Moreover, as the local parts may differ in their nature from each other, as well as from the global part, we seek a model rich enough to cover such a variety of scenarios. In practice, this also necessitates an easily-modifiable and scalable implementation. Last but not least, as there also exist situations where we cannot guess a-priori which parts of the data are global and which are local, we need a reliable and practical mechanism for automatically inferring that distinction.

Our key contributions are as follows. 1) To address the general setting above, we propose the Versatile HDPMM (vHDPMM), a novel Bayesian Nonparametric mixture model which is more flexible than HDPMM. While in HDPMM points are drawn solely from global (*i.e.*, shared) components, in vHDPMM one part of each point is drawn from a global component while its other part is drawn from a local component tied (in a many-to-one manner) to the global one. The model allows for differences (between the global and local parts and between local parts in different groups) in dimensions, data types (*e.g.*, continuous vs. discrete), and distributions. 2) Backed by a new theorem, we provide an effective and scalable mechanism for identifying global parts. 3) An appropriate new parallel sampling-based inference algorithm. The proposed sampler draws inspiration from an existing HDPMM parallel sampler [11] but differs from it by not only the fact it addresses a more general setting but also its superior scalability. Thus, even when the vHDPMM setting is reduced to that of HDPMM (*i.e.*, no local features), the proposed sampler scales better. 4) An efficient distributed and easily-modifiable implementation that supports various types of distributions and, in the reduced setting, is complementary to HDPMM implementations that are specialized to (a specific form of) topic modeling. Our Julia implementation is available at `www.github.com/BGU-CS-VIL/VersatileHDPMixtureModels.jl`, with an optional python wrapper at `www.github.com/BGU-CS-VIL/VersatileHDPMixtureModels`.

2 RELATED WORK

The Hierarchical Dirichlet Process (HDP) and HDPMM [39] extend the Dirichlet Process (DP) [13] and the Dirichlet Process Mixture Model (DPMM) [2, 46], respectively. Applications of HDP/HDPMM include, but are not limited to: a prior over Hidden Markov Models (HMMs) [15, 16], topic modeling and natural language processing [40, 28], tracking [14], computer vision [37], and music [25]. Since their inception, many HDP/HDPMM variants were proposed. While we are unaware of hierarchical models (nested or tree-like included) that address our full setting, some of those variants target problems close to ours. In [35] the data consists of 2 global parts (*i.e.*, no local), each is modeled via an HDPMM, and the two HDPMMs are coupled. In [34, 38], the prior is group-specific while in [23, 30, 7, 21, 48] various types of hierarchical data are modeled. The model in [38] addresses a global-local structure but focuses on Gaussian processes and does not tie local components to global ones.

Developing efficient HDPMM inference is an active research field where the two main paradigms are variational methods (*e.g.*, [45, 41, 44, 4, 26]) and sampling. Our proposed inference belongs to the latter. Recent samplers include those relying on particles [3], splits and

Table 1: Comparison with key HDPMM Samplers

	[43]	[11]	[31]	[22]	[5]	Ours
Does not require Stirling #s.	✓	×	×	×	✓	✓
Splits and merges	✓	✓	✓	×	×	✓
Parallel Sampling	×	✓	✓	✓	×	✓
Distributed Implementation	×	×	✓	✓	×	✓
Publicly-available code	✓	✓	×	×	✓	✓
Code supports not only categorical components	×	×	-	-	×	✓

merges [27, 43, 11] and/or parallel and/or distributed sampling [15, 36, 22, 47, 31, 20, 11]. *One thing that hinders the scalability of several HDPMM samplers is their reliance on Stirling numbers of the first kind*; our method circumvents that. Practical publicly-available implementations of HDPMM samplers are restricted to a specific form of topic modeling (*i.e.*, they support only categorical components); our implementation is more flexible and supports components from any exponential family. A comparison of several key HDPMM samplers with the proposed vHDPMM sampler (when the latter is reduced to the setting of HDPMM) is summarized in Table 1.

3 METHOD

3.1 THE PROPOSED MODEL

As in [39], in our Bayesian nonparametric hierarchical mixture model each group j has its own mixture weights, $\boldsymbol{\pi}_j$. While in [39] every mixture component, in its entirety, is global (*i.e.*, shared by all groups) our model is more flexible: each mixture component has both a global part and a local (*i.e.*, group-specific) part which is an entire mixture by itself. Below are the details, while Fig. 2 shows an associated graphical model. Let $\boldsymbol{x}_j = (x_{ji})_{i=1}^{n_j}$. We model $\boldsymbol{x} = \bigcup_{j=1}^{J} \boldsymbol{x}_j$ by a DPMM with a concentration parameter γ and an either continuous or discrete base measure H (note this is already slightly different from the HDPMM; see **Sup. Mat.** for details). Let $G_0 \sim \mathrm{DP}(\gamma, H)$ denote a discrete random measure drawn from that DPMM, and let $\boldsymbol{\beta} = (\beta_k)_{k=1}^{\infty}$ denote its atoms. Let $(G_j)_{j=1}^{J}$ denote discrete random measures drawn from a DPMM with a concentration parameter α and a base measure G_0. Let $\boldsymbol{\pi}_j = (\pi_{jk})_{k=1}^{\infty}$ denote the weights associated with G_j. Since a DPMM entertains the notion of a mixture of infinitely-many components, denoted by $\boldsymbol{\theta} = (\theta_k)_{k=1}^{\infty}$ – these are the atoms of G_j (hence also of G_0) – we write

$$p(\boldsymbol{x}_j|\boldsymbol{\theta}, \boldsymbol{\pi}_j) = \prod_{i=1}^{n_j} \sum_{k=1}^{\infty} \pi_{jk} f(x_{ji}; \theta_k),$$
$$\pi_{jk} > 0 \,\forall k, \qquad \sum_{k=1}^{\infty} \pi_{jk} = 1\,, \tag{1}$$

where f is a group-independent probability distribution

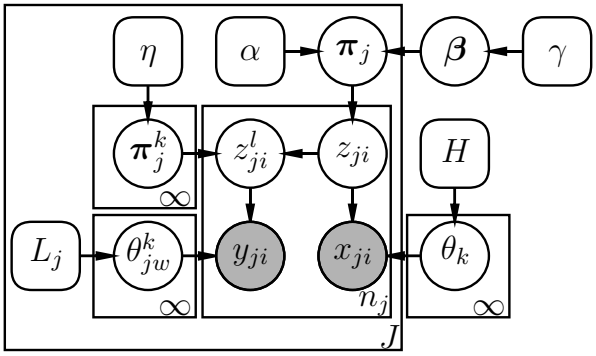

Figure 2: The vHDPMM as a graphical model.

function (pdf) or probability mass function (pmf) parameterized by θ_k and where $(\theta_k)_{k=1}^{\infty}$ and $\boldsymbol{\pi}_j$ themselves are drawn from their own prior distributions. The weights, $\boldsymbol{\pi}_j$, are drawn using a GEM stick-breaking process [33] with a concentration parameter $\alpha > 0$. The parameters, $(\theta_k)_{k=1}^{\infty}$, are *i.i.d.* draws from their prior, H. We make no distinction between $f(\cdot;\theta_k)$ and θ_k, referring to both as global component k. Using hidden labels, $\boldsymbol{z}_j = (z_{j1}, \ldots, z_{jn_j})$ with $z_{ji} = k$ if and only if x_{ji} is drawn from global component k, we rewrite Eq. (1) as

$$z_{ji} \overset{i.i.d.}{\sim} \mathrm{Cat}(\boldsymbol{\pi}_j) \quad i = 1, \ldots, n_j,$$
$$p(\boldsymbol{x}_j|\boldsymbol{\theta}, \boldsymbol{z}_j) = \prod_{i=1}^{n_j} f(x_{ji}; \theta_{z_{ji}})\,. \tag{2}$$

We define *global cluster* k as

$$c_k = (x_{ji})_{z_{ji}=k, j\in(1,\ldots,J),\, i\in(1,\ldots,n_j)}\,. \tag{3}$$

Let K, a latent random variable, be the number of global clusters; note that $K \leq \sum_{j=1}^{J} n_j$. By possibly renaming indices, we may assume without loss of generality that $\bigcup_{j=1}^{J} \boldsymbol{z}_j = \{1, 2, \ldots, K\}$. Within each group j, we also have a local DPMM, $\mathrm{DP}(\eta, L_j)$ where η is its concentration parameter and the group-specific local base measure L_j is either continuous or discrete. For each $j \in \{1, \ldots, J\}$ and each $k \in \{1, \ldots, K\}$, let

$$s_j^k = (y_{ji})_{i:z_{ji}=k}\,. \tag{4}$$

To each s_j^k we attach its own ∞-component mixture, drawn from $\mathrm{DP}(\eta, L_j)$. We denote its atoms by $\boldsymbol{\theta}_j^k = (\theta_{jw}^k)_{w=1}^{\infty}$ and its weights by $\boldsymbol{\pi}_j^k = (\pi_{jw}^k)_{w=1}^{\infty}$:

$$p(s_j^k|\boldsymbol{\theta}_j^k, \boldsymbol{\pi}_j^k) = \prod_{i:z_{ji}=k} \sum_{w=1}^{\infty} \pi_{jw}^k f_j(y_{ji}; \theta_{jw}^k),$$
$$\pi_{jw}^k > 0 \,\forall w, \quad \sum_{w=1}^{\infty} \pi_{jw}^k = 1\,. \tag{5}$$

Even within each group j, the local weights, $\boldsymbol{\pi}_j^k$, are not shared across $(s_j^k)_{k=1}^{\infty}$. If L_j is not discrete, the same holds for the local atoms, $\boldsymbol{\theta}_j^k$. Likewise, $(\boldsymbol{\pi}_j^k)_{k=1}^{\infty}$ are not shared across the groups. This is also the case for $(\boldsymbol{\theta}_j^k)_{k=1}^{\infty}$, unless $(L_j)_{j=1}^{J}$ are all discrete *and* share their

atoms. We now write, using hidden local labels,

$$z_{ji}^l \overset{i.i.d.}{\sim} \text{Cat}(\boldsymbol{\pi}_j^k) \quad \forall i \text{ s.t. } z_{ji} = k$$
$$p(s_j^k | \boldsymbol{\theta}_j^k, \boldsymbol{z}_j^l) = \prod\nolimits_{i:z_{ji}=k} f_j(y_{ji}; \theta_{z_{ji}^l}^k) \tag{6}$$

where f_j is a local (*i.e.*, group-specific) pdf or pmf parameterized by θ_{jw}^k, $\boldsymbol{\pi}_j^k$ is drawn from a GEM [33] stick-breaking process with a concentration parameter η, $(\theta_{jw}^k)_{w=1}^{\infty}$ are *i.i.d.* draws from their prior, L_j, $z_{ji}^l = w$ if and only if y_{ji} is drawn from local component θ_{jw}^k, and $\boldsymbol{z}_j^l = (z_{ji}^l)_{i=1}^{n_j}$. We define *local cluster* w, in group j and tied (usually in a many-to-one manner) to c_k, as

$$c_{jw}^k = (y_{ji})_{i:(z_{ji}, z_{ji}^l)=(k,w)} \,. \tag{7}$$

Let K_j^k denote the latent random number of local clusters in group j tied to global cluster k. Thus, $s_j^k = (c_{jw}^k)_{w=1}^{K_j^k}$.

Back to the cosegmentation example, let H be a Normal-Inverse-Wishart (NIW) prior in 3D and let L_j be an NIW prior in 2D. Let f be a 3D Gaussian pdf with a 3D mean and a 3-by-3 covariance, $\theta_k = (\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$, and let f_j be a 2D Gaussian pdf with a 2D mean and a 2-by-2 covariance, $\theta_{jw}^k = (\boldsymbol{\mu}_{jw}^k, \boldsymbol{\Sigma}_{jw}^k)$. The color Gaussians are shared across the images but the spatial ones are not.

More generally, the local parts may differ from each other (and from the global part) in their distributions, types (*e.g.*, continuous vs. discrete), and dimensions (including the case where some or all of the local dimensions are zero; *i.e.*, no local features). The setting of vHDPMM generalizes that of HDPMM: the latter is a particular case of the former when all the local dimensions are zero. We now stretch the Chinese Restaurant Franchise (CRF) metaphor [39], adapting it to vHDPMM. In our case, a franchise has J restaurants of infinitely-many tables. Customer i entering restaurant j sits at table t_{ji}. On each table in restaurant j there is both a franchise-owned dish (offered by all J restaurants) and a restaurant-specific side dish, where, in that restaurant, the same dish may appear on more than one table but only provided it is accompanied by a different side dish at each time. In other words, whenever different tables have the same pair of dish and side dish, we unite ("collapse") these tables to a single one. As a particular case, in the reduced setting when there are no side dishes, in each of our restaurants, tables with the same dish are collapsed to a single table (unlike in HDPMM where different tables at the same restaurant may offer the same dish). The franchise dish menu and each of the restaurant-specific side-dish menu are of infinite lengths. Our CRF is more welcoming than in [39] whose restaurants offer no side dishes. Also, even in the reduced HDPMM setting (no side dishes), there is a subtle difference from HDPMM. In vHDPMM, if a customer entering a restaurant decides to open a new table, the dish on that table is drawn according to how many customers (in the franchise) sit next to a table with that dish; this is unlike in [39], where the dish is drawn according to how many tables (in the franchise) have it. We will return to this key point later.

Algorithm 1: A single iteration of the proposed parallel vHDPMM Sampler

```
Draw parameters and weights θ, β, θ̄, β̄ by Eqs. (8) and (9)
for j ∈ (1, ..., J) do in parallel
    Draw parameters and weights
      π_j, θ_jw^k, π_j^k, θ̄_jw^k, π̄_j^k   by Eqs. (10)-(12)
    for i ∈ (1, ..., n_j) do in parallel
        Draw assignments z_ji, z̄_ji, z_ji^l, z̄_ji^l by
          Eqs. (13)
        Calculate sufficient statistics
    for k ∈ (1, ..., K) do in parallel
        for for each local cluster c_jw^k do in parallel
            Propose and accept/reject a local split
        for for each pair of local clusters c_jw^k and
          c_jw'^k do in parallel
            Propose and accept/reject a local merge
              (avoid clashes as in [10])
Propose and accept/reject global splits/merges
```

3.2 THE PROPOSED SAMPLER

The HDPMM CRF-based sampler [39] scales poorly due to its serial nature; moreover, making only small moves, it is highly susceptible to poor local maxima. As an analogous vHDPMM CRF-based sampler (included in our **Sup. Mat.**) has similar drawbacks, *a better vHDPMM sampler is needed*. One HDPMM sampler superior to the CRF-based sampler was proposed by Chang and Fisher [11]; it extends their DPMM sampler [10]. Both their samplers use parallel sampling, model augmentation via sub-clusters, and splits/merges. Our proposed parallel vHDPMM sampler, summarized in Algorithm 1, is inspired by [10, 11] but differs from both in 3 key aspects: **1)** It addresses challenges posed by the new model; *i.e.*, it accommodates both global and local features and performs splits/merges for both local and global components. **2)** While its local splits/merges are similar to those in [10], its global splits/merges are accepted or rejected differently from those in [11]. **3)** It scales better, w.r.t. the number of points per group, than the one from [11].

Augmenting the vHDPMM. The augmentation, visualized in the **Sup. Mat.**, is as follows. To each component, local or global, we attach an auxiliary 2-component mixture. Global cluster c_k is already associated with a global

component, θ_k, and a global weight, β_k. In addition, we now also associate c_k with 2 global sub-*components*, $\bar{\theta}_k = (\bar{\theta}^1_k, \bar{\theta}^2_k)$, with corresponding weights, $\bar{\beta}_k = (\bar{\beta}^1_k, \bar{\beta}^2_k)$. Together, $\bar{\theta}_k$ and $\bar{\beta}_k$ are used to partition c_k into 2 global sub-*clusters*, $(\bar{c}^1_k, \bar{c}^2_k)$. Similarly, for each $w \in (1, \ldots, K^k_j)$, we define 2 local sub-*components*, $\bar{\theta}^k_{jw} = (\bar{\theta}^{k,1}_{jw}, \bar{\theta}^{k,2}_{jw})$, with corresponding weights, $\bar{\pi}^k_{jw} = (\bar{\pi}^{k,1}_{jw}, \bar{\pi}^{k,2}_{jw})$, to be used for partitioning local cluster c^k_{jw} into 2 local sub-*clusters*, $(\bar{c}^{k,1}_{jw}, \bar{c}^{k,2}_{jw})$. For each global label z_{ji} and local label z^l_{ji} we define $\bar{z}_{ji} \in \{1, 2\}$ and $\bar{z}^l_{ji} \in \{1, 2\}$; these are the global and local sub-cluster labels of point ji.

Similarly to [10, 11], we alternate between two steps: a restricted parallel Gibbs sampler (with fixed numbers of clusters) and splits/merges (for changing these numbers by utilizing the auxiliary variables). Both steps require specialization to vHDPMM as detailed below.

Restricted Parallel Gibbs Sampler. Our restricted sampler, adapted from [11], changes neither K nor any of the K^k_j values; rather, it can only modify the labels and existing components and weights. The restricted sampler (lines 1–6 in Algorithm 1), uses the following conditional distributions (whose detailed functional forms appear in the **Sup. Mat.** due to space limits):

$$p(\theta_k|c_k; H) \text{ and } p(\bar{\theta}^a_k|\bar{c}^a_k; H) \quad (a \in (1, 2)) \tag{8}$$

(for the global components and their sub-components);

$$p((\beta_k)^{K+1}_{k=1}|(c_k)^K_{k=1}; \gamma) \text{ and } p(\bar{\beta}_k|\bar{c}^1_k, \bar{c}^2_k; \gamma) \tag{9}$$

(for the corresponding weights);

$$p(\boldsymbol{\pi}_j|\boldsymbol{\beta}, (s^k_j)^K_{k=1}; \alpha) \tag{10}$$

(for the group-specific weights of the global components);

$$p(\theta^k_{jw}|c^k_{jw}; L_j)\,;\; p(\bar{\theta}^{k,b}_{jw}|\bar{c}^{k,b}_{jw}; L_j) \quad (b \in (1, 2)) \tag{11}$$

(for the local components and their sub-components);

$$p(\boldsymbol{\pi}^k_j|(c^k_{jw})^{K^k_j}_{w=1}; \eta) \text{ and } p(\bar{\pi}^k_{jw}|\bar{c}^{k,1}_{jw}, \bar{c}^{k,2}_{jw}; \eta) \tag{12}$$

(for the corresponding weights);

$$p(z_{ji}, z^l_{ji}|x_{ji}, y_{ji}, \boldsymbol{\pi}_j, (\boldsymbol{\pi}^k_j, \theta_k, (\theta^k_{jw})^{K^k_j}_{w=1})^K_{k=1}) \text{ and}$$
$$p(\bar{z}_{ji}, \bar{z}^l_{ji}|x_{ji}, z_{ji} = k, z^l_{ji} = w, \bar{\beta}_k, \bar{\pi}^k_{jw}, \bar{\theta}_k, \bar{\theta}^k_{jw}) \tag{13}$$

(for global and local labels).

An obvious difference between our restricted sampler and the one in [11] is that ours handles a more general setting (*i.e.*, the addition of local features). There is, however, another subtle difference: In Eq. (9), global weights are sampled not according to the counts of "tables" assigned to each component ("dish") as is done in [11]; rather, the counts stand for the number of observations ("customers") associated with the component; **While this detail might seem small, it has a drastic positive effect on the scalability**. For example, in the reduced setting of the vHDPMM, this detail lets our sampler scale better than that traditional HDPMM samplers ([11] included). We will return to this point and clarify it in § 3.3.

Splits and Merges. We use a Metropolis-Hastings framework [24] to propose two types of split/merges: global and local (not to be confused with the operations called global/local splits in [11] as those refer to something else). The Hastings ratios below assume conjugate priors.

Local Splits and Merges. A local split splits a local component into its two sub-components. A local merge merges two local components (and then the two become the sub-components of the new one). When we propose a split or a merge, we propose new values, denoted by $\hat{}$ instead of the current values. The proposal distributions, denoted by q, are as in [10] so we skip their details. For splits, we denote the current component by c, and the two sub-components of c by m and n. For merges, we denote the two proposed-to-be-merged components by m and n, and the resulting component by c. That is:

$$(\hat{c}^k_{jm}, \hat{c}^k_{jn}) = \text{split}(c^k_{jc}, \bar{c}^{k,1}_{jc}, \bar{c}^{k,2}_{jc}) \tag{14}$$
$$\hat{c}^k_{jc} = \text{merge}(c^k_{jm}, c^k_{jn}) \tag{15}$$
$$(\hat{\pi}^k_{jm}, \hat{\pi}^k_{jn}) = \pi^k_{jc} \times \bar{\pi}^k_{jc} \qquad \hat{\pi}^k_{jc} = \pi^k_{jm} + \pi^k_{jn} \tag{16}$$
$$(\hat{\theta}^k_{jm}, \hat{\theta}^k_{jn}) \sim q(\hat{\theta}^k_{jm}, \hat{\theta}^k_{jn}|\hat{c}^k_{jm}, \hat{c}^k_{jn}) \tag{17}$$
$$\hat{\theta}^k_{jc} \sim q(\hat{\theta}^k_{jc}|\hat{c}^k_{jc})\,. \tag{18}$$

The split function (Eq. (14)), splits c^k_{jc} into clusters $\hat{c}^k_{jm}, \hat{c}^k_{jn}$. To that aim it looks at the respective auxiliary labels and assigns x_{ji} to $\hat{c}^k_{jm}$ if $x_{ji} \in \bar{c}^{k,1}_{jc}$, and to $\hat{c}^k_{jn}$ otherwise. The merge function (Eq. (15)) merges clusters c^k_{jm}, c^k_{jn} into cluster $\hat{c}^k_{jc}$. *Local merges are allowed only for local clusters of the same global cluster k.* The resulting Hastings ratios (which, being related to a non-hierarchical DPMM, are based on those in [10]) are

$$H^{\text{M}}_{\text{l}} = \frac{\Gamma(|\hat{c}^k_{jc}|) f_j(\hat{c}^k_{jc}; L_j)}{\eta \prod_{a \in \{m,n\}} \Gamma(|c^k_{ja}|) f_j(c^k_{ja}; L_j)} \tag{19}$$

and $H^{\text{S}}_{\text{l}} = 1/H^{\text{M}}_{\text{l}}$ where the M and S stand for a merge and a split, respectively, and $\gamma(\cdot)$ is the gamma function.

Global Splits/Merges. The mechanism for proposing global splits/merges is similar to the one in [11], with modifications to accommodate the new model. In contrast to [11], when a global split is performed in our model *it affects multiple instances of that component in each group*, not just a single one. Moreover, the counts used are not of "tables" with the same "dish" (as is done in [11]) but of customers having the same dish (see also § 3.3). This implies a new Hastings ratio for global splits:

$$H_{\mathrm{g}}^{\mathrm{S}} = \frac{\gamma \prod_{a\in\{m,n\}} \Gamma(|\hat{c}_a|) f(\hat{c}_a; H)}{\Gamma(|c_c|) f(c_c; H)} \frac{{\pi_c}^{|\hat{c}_m|+|\hat{c}_n|}}{{\hat{\pi}_m}^{|\hat{c}_m|} {\hat{\pi}_n}^{|\hat{c}_n|}} \times \prod_{j=1}^{J} \prod_{w=1}^{K_j^c} \frac{\Gamma(\alpha\pi_c)}{\Gamma(\alpha\pi_c + |c_{jw}^c|)} \prod_{a=m,n} \frac{\Gamma(\alpha\hat{\pi}_a + |\hat{c}_{jw}^a|)}{\Gamma(\alpha\hat{\pi}_a)} . \quad (20)$$

A split of global cluster also splits all its local clusters in all the groups, often drastically increasing the overall number of local clusters. Merging global clusters $m, n \in \{1, \ldots, K\}$ changes, across all groups, the tying of all the local clusters (c_{jw}^m, c_{jw}^n) from either m *or* n to c. Unlike in the local-move case, and unlike in [11], *our global splits and merges are not the inverse of each other*, due to the local-cluster count remaining constant in a global merge; likewise, their Hastings ratios are **not** reciprocal. This yields a new Hastings ratio for a global merge:

$$H_{\mathrm{g}}^{\mathrm{M}} = \frac{\Gamma(|\hat{c}_c|) f(\hat{c}_c; H)}{\gamma \prod_{a\in\{m,n\}} \Gamma(|c_a|) f(c_a; H)} \frac{{\hat{\pi}_c}^{|c_m|+|c_n|}}{{\pi_m}^{|c_m|} {\pi_n}^{|c_n|}} . \quad (21)$$

3.3 SCALABILITY

In § 3.4 we discuss our distributed implementation that can use multiple multicore machines. However, even on a single machine and even in the reduced setting, our sampler scales better than [11] and other HDPMM samplers when the n_j increases. This is partly due to the following. Recent samplers [11, 31, 22] use the Direct Assignment (DA) construction [40] which relies on pre-computing Stirling numbers of the 1st kind (not to be confused with the Generalized Stirling numbers used in [5]) for estimating the number of *tables* in each restaurant. Such Stirling numbers are computed recursively and cost in both runtime and memory. While applicable for the common domain of topic modeling, where the n_j's are relatively small (short documents), this is impractical for larger groups (*e.g.*, on our machine, for $n_j = 10^6$ these take 24hr to compute or a 4TB lookup table). In contrast, since our model gives rise to a sampler that never needs to estimate the table counts, our sampler easily handles domains such as image cosegmentation, where each image is over 10^6 pixels (so DA-based samplers are inapplicable).

3.4 IMPLEMENTATION

Unlike the implementations from [10, 11] and more like the one from [12] (a reimplementation of the non-hierarchical sampler from [10]), our proposed implementation is not only parallel but also distributed, and can thus utilize multiple machines. In this section, the term "process" refers to a computer process, not a stochastic process. Algorithm 1 describes a single iteration of the parallel sampler. Due to inter-group conditional independence, group-specific computations can be parallelized using different processes, including processes on different machines. For each group j, an intra-group parallelism can be be exploited. We thus implemented, in Julia, a distributed parallel sampler which exploits these facts. Its details, together with a runtime analysis, appear in the **Sup. Mat**. We chose Julia since it offers a good combination of abstraction, high performance, a fairly-painless way to distribute computations, and ease of code modifications. A key feature of our implementation is that it supports any exponential family of distributions for either the global or local parts. Unlike most HDPMM implementations (*e.g.*, see Table 1) it is not limited to categorical components (which are often used in topic modeling) and thus, *e.g.*, also supports Gaussians, mulitinomials, etc.

3.5 DETECTING GLOBAL FEATURES

Unlike in the cosegmentation example, it is not always clear which features are global and which are not. We now show how to automate that decision. Suppose that there indeed exists a latent distinction between global and local parts. If some features exist in only a subset of the groups, they must be local. Thus, without loss of generality, assume that all the features appear in all groups, implying all points have a shared dimension, denoted by D. Let D_g and D_l be the dimensions of the global and local features, respectively; *i.e.*, $D_g + D_l = D$. Assume first that D_g and D_l are known, an assumption we will relax later. Let $L = (L_1, \ldots, L_J)$ be the local base measures, let $\boldsymbol{R} = (\boldsymbol{R}_1, \ldots \boldsymbol{R}_J)$ denote the data where each $\boldsymbol{R}_j$ is a D-by-n_j matrix with columns $(r_{ji})_{i=1}^{n_j}$, and let $\boldsymbol{\Pi}$ be a D-by-D permutation matrix. Define an *optimal feature partitioning* by maximizing the (marginal) likelihood:

$$\arg\max_{\boldsymbol{\Pi}} p(\boldsymbol{\Pi}\boldsymbol{R} | \mathrm{vHDPMM}(H, \gamma, \alpha, L, \eta)) \quad (22)$$

where the first D_g rows of $\boldsymbol{\Pi}\boldsymbol{R}$ are regarded as the global features (note that it is implicitly understood that H and L depend on which features are defined, by $\boldsymbol{\Pi}$, as global or local). *Since the proposed implementation is very fast, we can find* $\boldsymbol{\Pi}$ *via an exhaustive search as long as* D *is not too high* (*e.g.*, 1000-dimensional Gaussian components are still acceptable even on a single 4-core machine; see § 4). This is especially true since we can opt to run the search on only a random subset of the data and since it takes only few iterations to asses a nominal $\boldsymbol{\Pi}$ (there is no need to wait for convergence). The theorem below addresses the more interesting case where D_g and D_l are unknown; for a proof, see our **Sup. Mat.**

Theorem 1 (Suboptimal Feature Partitioning) *Let* $D'_g > 0$ *and* $D'_l > 0$ *be integers summing to* D*, with* $D'_g \leq D_g$*. Suppose that we know that certain* D'_g *features are global. Let* $\boldsymbol{R}'$ *be obtained by some*

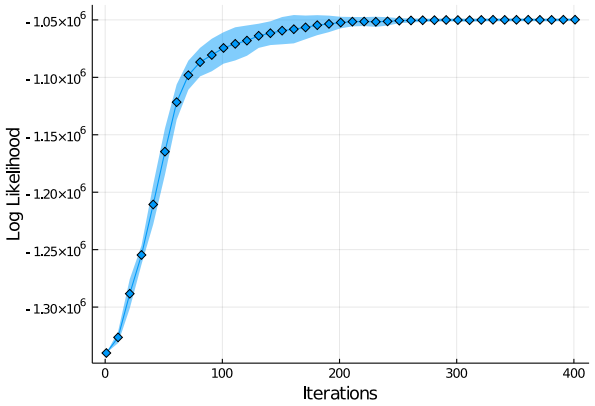

Figure 3: A typical convergence of the sampler. Data: 10 groups; 20K per group; 50 global 5-dimensional Gaussians; 200 local 5-dimensional Gaussians in each group. The log-likelihood results shown are the mean (line) and std. dev. (shaded area) over 10 runs of the sampler.

permutation of the rows of $\boldsymbol{R}$ *such that the first* D'_g *rows of* $\boldsymbol{R}$ *are these* D'_g *global features. Let* $\boldsymbol{R}''$ *denote the matrix obtained by copying the first* D'_g *rows of* $\boldsymbol{R}'$*, and permuting its remaining rows such that in row* $(D'_g + 1)$ *of* $\boldsymbol{R}''$ *we have one of the* D_l *local features. Let* $\mathrm{vHDPMM}(H', \gamma, \alpha, L', \eta)$ *denote the vHDPMM model that treats the first* D'_g *rows of* $\boldsymbol{R}'$ *as global features and let* $\mathrm{vHDPMM}(H'', \gamma, \alpha, L'', \eta)$ *denote the vHDPMM model that treats the first* $D'_g + 1$ *rows of* $\boldsymbol{R}''$ *as global features where* H'*,* H''*,* L'*, and* L'' *are base measures (of the proper dimensions) analogous to* H *and* L*. Then,*

$$p(\boldsymbol{R}'|\,\mathrm{vHDPMM}(H', \gamma, \alpha, L', \eta)) > p(\boldsymbol{R}''|\,\mathrm{vHDPMM}(H'', \gamma, \alpha, L'', \eta))\,. \quad (23)$$

That is, if we already correctly found that one or more features are global, then when mistakenly adding a local feature as another global feature, the likelihood will drop.

For some intuition, see Fig. 3 and Fig. 4 in the **Sup. Mat.** As those figures show, mistaking a local feature to be global yields a noticeably-poor clustering. *More generally, the likelihood will in fact usually increase (and will never decrease) as we correctly identify more and more global features as such, as long as we do not mistakenly declare local features as global – but, by the theorem, such a mistake is detected via a likelihood drop*. As the theorem does not apply to $D'_g = 0$, the first global feature is the most challenging to find. However, that too can be done using an exhaustive search where again we can use a random subset of the data (*e.g.*, 10%) and a small number of iterations (*e.g.*, 20). Once the first global feature is found this way, Theorem 1 provides a justification to a simple fully-automated mechanism that lets us detect additional global features in reasonable times even in fairly-high dimensions. That is, we gradually try increasing the set of features that are declared as global. If the likelihood drops, we declare the feature in question as local; otherwise, it is global. When running the mechanism we can use a subset of the data (Theorem 1 holds for not only the entire data but also any of its subset), and a very small number (*e.g.*, 20) of iterations to detect a possible likelihood drop. Also useful is the fact that once enough global features are detected, identifying the rest is easier; *i.e.*, the likelihood drop when adding a local feature as global is noticeable within a smaller number of iterations (*e.g.*, 10). Moreover, Theorem 1 holds for any $D^* < D$. Thus, if we have, say, $D = 1010$, after finding the first 10 global features, we partition the remaining features to data chunks of 100 dimensions each. To each chunk, we append the already-found 10 global features, making the data in it of dimension $D^* = 110 < D = 1010$. We then run the mechanism on each chunk. Finally, it is fine to miss some global features: as long as we correctly find enough of them, the clustering will usually succeed; it is only important to not misclassify local features as global, but such mistakes are easily detected (by Theorem 1) and thus avoided. To summarize, *the mechanism scales well and is practical even in fairly-high dimensions*; see § 4.

4 RESULTS

We tested the proposed method on synthetic data as well as on real-data applications such as image cosegmentation, visual topic modeling, and clustering with missing data. Details omitted here (due to page limit) about machine specs, measured running times, and hyperparameter values, can be found in the **Sup. Mat.**

Empirical convergence. The theoretical guarantees for the convergence of the sampler are similar to the samplers in [10, 11]. Empirically, the sampler indeed consistently converges as is demonstrated in Fig. 3.

Inference when the "the model is right". We generated 6 synthetic datasets (whose names are explained below): $\mathcal{G}3/2/5/1$; $\mathcal{G}5/2/10/1$; $\mathcal{G}3/5/5/5$; $\mathcal{G}5/5/10/5$; $\mathcal{M}10/5/20/5$; $\mathcal{M}10/200/20/200$. Each one contained 10 groups with 20K points per group. In the first 4 datasets, we used Gaussian components at both the global and local levels, drawn from NIW priors. In the last 2, we used Multinomials, again at both levels, drawn from Dirichlet-*distribution* priors. Particularly, in each dataset we drew D_g-dimensional features from a global K-component mixture and then, in each group j and each global component k, we drew D_l-dimensional local features from a K'-component local mixture. The naming convention is $K/D_g/K'/D_l$; *e.g.*, in $\mathcal{G}3/2/5/1$ we used a 3-component Gaussian Mixture Model (GMM) over

Table 2: NMI Scores on synthetic data (10 groups, 20K points per group, 10 runs per model)

	$\mathcal{G}5/2/10/1$		$\mathcal{G}5/5/10/5$	
Method	G	L	G	L
HDPMM-all	0.705	-	0.632	-
HDPMM-global-only	0.769	-	0.888	-
DPMM-merged	-	0.076	-	0.193
DPMM-separated	-	0.667	-	0.903
vHDPMM (ours)	**0.862**	0.746	**0.959**	0.872
vHDPMM-seperated (ours)	-	**0.751**	-	**0.972**

2D global features and a 5-component GMM over 1D local features. As a performance index, we computed the mean Normalized Mutual Information (NMI) for 6 models: **HDPMM-all** (an HDPMM when using all the features); **HDPMM-global-only** (an HDPMM when using only the global features); **DPMM-merged** (a single DPMM when the known partition to groups is discarded); **DPMM-separated** (J independent DPMMs, one in each group); the proposed **vHDPMM**; **vHDPMM-separated** (J independent vHDPMMs, one in each group). These mean NMIs were computed over 10 runs (the standard deviations were all < 0.01). For each dataset there are two columns in Table 2, denoted by G (global clusters) or L (local). In each run, DPMM-separated's NMI score was obtained by averaging scores of the per-group DPMMs. The vHDPMM-separated's score was similarly calculated. The results on $\mathcal{G}5/2/10/1$ and $\mathcal{G}5/5/10/5$ appear at Table 2 while the results on the other 4 datasets are in the **Sup. Mat.** The results are unsurprising. Locally, the vHDPMM-separated wins and the DPMM-separated is the runner up (vHDPMM being usually the third); the vDPMM-separated and vHDPMM-separated, however, cannot make global inference (*e.g.*, persistence across groups). Globally, vHDPMM is the best. This is because rather than discarding the local features (as the HDPMM-global-only does), or getting confounded by them (as happens to the HDPMM-all), *the vHDPMM leverages local information to improve global clustering*. The main reason we used NMI, a popular clustering measure, is that it is model independent. Measures such as the posterior or likelihood are model dependent so comparing them across different models is uninformative.

Image cosegmentation: vHDPMM in its full setting (real data). In the first cosegmentation experiment, our 5D feature vector in each pixel consisted of the 3D color and 2D location. We used Gaussian components and NIW base measures. The first dataset consistent of 5 images with $\sim 1.6 \cdot 10^6$ pixels each. First, we ran our mechanism for global-feature detection. It deemed, within seconds, that only the colors are global. Thus, we ran our vHDPMM sampler with color and location as global and local features, respectively. Example results are shown in Fig. 1 while the full results are in the **Sup. Mat.** The second dataset consisted of 50 frames from [6]. Here, rather than working on pixels, we first computed superpixels (using [42]), primarily since they yielded more visually-pleasing results. From each superpixel we extracted its mean location-color value, and ran the vHDPMM sampler on those values. See **Sup. Mat.** for results.

Table 3: NMI Scores for Clustering With Missing Data

Method	**NMI (Averages of 50 Runs)**
GMM (saw all data)	0.72±0.014
DPGMM (saw all data)	0.743±0.052
vHDPMM (all data)	0.821±0.043
vHDPMM (missing data)	0.804±0.05

Clustering with missing data (real data). In this experiment we used a 9-class Tetragonula Bees classification data [17] of 236 13D samples. We fitted a 9-component GMM [32] and a DPGMM [12] to the data. Next, we divided at random the data into 4 equal-size groups. In each group we kept the first 6 features, declared them as global, and, to simulate missing data, removed a group-specific random number of the other 7 features, declaring the remaining ones as local. We ran vHDPMM on this new dataset. Thus, the vHDPMM did not see all the features, and its local parts varied in dimensions (between groups, and between different runs). In addition, we also ran vHDPMM when all 7 local features were present in all groups. Table 3 shows that vHDPMM beats the DPGMM and GMM regardless if it saw all the data (like them) or not. These surprising results (which the reader is welcome to reproduce by running our code) can be possibly explained by the fact that if a feature is "bad" for clustering, there is an advantage to treating it as local.

We now shift the discussion to experiments where the setting of vHDPMM is reduced to that of HDPMM. Beyond the specific case of topic modeling with categorical components (as opposed to topic modeling via multinomial components, or cases outside topic modeling), available implementations of HDPMM inference are hard to find (we also note that many DPMM papers mention they tested extensions to HDPMM, but usually do no release code for that). Moreover, many HDPMM methods, [11] included, scale poorly with the groups' sizes (see § 3.3), making then inapplicable in various cases where using HDPMM would be otherwise natural.

Gaussian Components. In this experiment we tested our sampler on data generated from a CRF prior over an infinite mixture of Gaussian components in varying dimensions and sample counts. We compared with a baseline of a CRF-based HDPMM sampler. Table 4 shows that our sampler yields better results and is orders of magnitude faster. The last configuration, with $n_j = 50K$, is

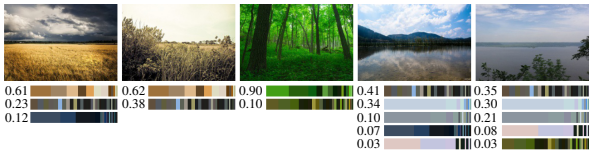

Figure 4: Visual Topic Modeling. The sampler inferred 11 shared visual topics from 100 landscape images. Five example images are shown, each with its inferred topic mixture, where only the top topics in each topic are displayed (cumsum ≥ 0.95). Each topic is a color histogram, visualized here as a color stripe (segment's length = bin's height).

Table 4: Inference in data sampled from an HDPMM-CRF prior over an infinite mixture of D-dimensional Gaussian components, 4 groups, N points per group, 3 runs. We ran our sampler (in the reduced HDPMM setting) and a CRF-based HDPMM sampler until convergence.

	HDPMM-CRF		**Proposed Sampler**	
Data	*NMI*	*Time [sec]*	*NMI*	*Time [sec]*
D:3, N:100	0.79±0.03	80±0.9	**0.81**±0.01	**1.56** ±0.2
D:8, N:5K	**0.86**±0.02	6760±93	**0.86**±0.01	**18.6** ±1.3
D:15, N:50K	0.91±0.02	68K±2K	**0.96**±0.006	**179** ±13

beyond the scope of methods using those Stirling numbers (not to mention their lack of support for Gaussians).

Visual topic modeling of 100 landscape images using multinomial components (real data). This experiment is beyond the scope of existing available HDPMM implementations as it involves: 1) multinomials; 2) large groups. Concretely, each image (of varying sizes) was considered a group and was partitioned into 2×2 patches. From each patch we extracted an HSV 256-bin sparse histogram. Thus our data was 100 groups (images) with $10K - 15K$ samples in each group, and each sample was a 256-bin histogram. Each learned visual topic is represented by its 8 most probable colors, and each image has its own mixture of such shared visual topics. See Fig. 4.

Image cosegmentation: vHDPMM in its reduced setting (real data). Here we repeated the cosegmentation experiments described above (again using pixels for the first dataset and superpixels for the second) but this time we used only RGB values, to obtain the reduced setting

Table 5: Running time versus the number of machines (each with 12 processes) on 50 million 5D data points (divided into 200 groups). See text for more details.

Number of Machines	1	2	3	4
Time [sec]	3299	2552	2107	1862

Table 6: Detecting global features via an exhaustive search in the case of high-dimensional Gaussians. Data: 4 groups, 10^5 samples per group, sampled from 10 global clusters and 20 local clusters (in each group), $Gdim = Ldim = D/2$. TP=True Positives; FP=False Positives; TN=True Negatives; FN=False Negatives. We used a single 4-core machine (Intel i5-6600 CPU @ 3.30GHz).

D	**TP**	**FP**	**TN**	**FN**	**Time [sec]**
100	43	0	50	7	92
500	232	0	250	18	446
1000	474	0	500	26	889

of the HDPMM. To emphasize scalability of our sampler, note that even though in the first dataset the number of pixels in each image was $n_j = 1606124$ (beyond the scope of other HDPMM methods), our sampler converged successfully within 210 [sec]. The results appear in the **Sup. Mat.** On both these datasets, a comparison between the full and reduced setting show that the local information, unusable by HDPMM, improves the results; *e.g.*, in the HDPMM setting, small components are often not captured and similar colors are poorly separated.

Running Time. To test the effect the number of machines has on the running time, we took 200 consecutive video frames (from [29]), each with 250K pixels, and ran on either 1,2,3 or 4 multicore machines our vHDPMM sampler (with color as global features and location as local) till convergence. In each of the runs the sampler inferred 11 global clusters, and about 5000 local clusters. The timing results in Table 5 show the scalability of our method.

Scalability of the mechanism for detecting global features. Table 6 shows that the mechanism scales gracefully with the dimension.

5 CONCLUSION

We proposed a novel framework for clustering grouped data in settings more general than those of HDPMM. The approach is scalable and flexible, and its effectiveness was shown on synthetic and real datasets. In the reduced HDPMM setting, it scales better than HDPMM methods w.r.t. the number of points per group. Our work thus facilitates the usage of HDPMM-related ideas in new domains where it used to be infeasible. We also provided a new theoretical result that helps discover global features. While this paper was driven by clustering and applications and thus focused on efficient parallel sampling and its distributed implementation, future work may explore theoretical characterizations of the new model such as the expected number of global and local clusters, and estimating convergence rates.

References

[1] Achanta et al. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE TPAMI*, 2012.

[2] Antoniak. Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. *AoS*, 1974.

[3] Bouchard et al. Particle Gibbs split-merge sampling for Bayesian inference in mixture models. *JMLR*, 2017.

[4] Bryant and Sudderth. Truly nonparametric online variational inference for hierarchical Dirichlet processes. In *NIPS*, 2012.

[5] Burkhardt and Kramer. Online sparse collapsed hybrid variational-gibbs algorithm for hierarchical Dirichlet process topic models. In *ECML-PKDD*, 2017.

[6] Butler et al. A naturalistic open source movie for optical flow evaluation. In *ECCV*, 2012.

[7] Canini and Griffiths. A nonparametric Bayesian model of multi-level category learning. In *AAAI*, 2011.

[8] Carson et al. Blobworld: A system for region-based image indexing and retrieval. In *ICAVIS*. Springer, 1999.

[9] Chang et al. A video representation using temporal superpixels. In *CVPR*, 2013.

[10] Chang and F. III. Parallel sampling of DP mixture models using sub-cluster splits. In *NIPS*, 2013.

[11] Chang and F. III. Parallel sampling of HDPs using sub-cluster splits. In *NIPS*, 2014.

[12] Dinari et al. Distributed MCMC inference in Dirichlet process mixture models using Julia. In *CCGRID HPML Workshop*, 2019.

[13] Ferguson. A Bayesian analysis of some nonparametric problems. *AoS*, 1973.

[14] Fox et al. Hierarchical Dirichlet processes for tracking maneuvering targets. In *ICDF*. IEEE, 2007.

[15] Fox et al. An HDP-HMM for systems with state persistence. In *ICML*. ACM, 2008.

[16] Fox et al. A sticky HDP-HMM with application to speaker diarization. *AoAS*, 2011.

[17] Franck et al. Nest architecture and genetic differentiation in a species complex of Australian stingless bees. *Molecular Ecology*, 2004.

[18] Freifeld et al. A Fast Method for Inferring High-Quality Simply-Connected Superpixels. In *ICIP*, 2015.

[19] O. Freifeld, H. Greenspan, and J. Goldberger. Multiple sclerosis lesion detection using constrained GMM and curve evolution. *IJBI*, 2009.

[20] Gal and Ghahramani. Pitfalls in the use of parallel inference for the Dirichlet process. In *ICML*, 2014.

[21] Gao et al. Tracking and connecting topics via incremental hierarchical Dirichlet processes. In *ICDM*. IEEE, 2011.

[22] Ge et al. Distributed inference for Dirichlet process mixture models. In *ICML*, 2015.

[23] Griffiths et al. Hierarchical topic models and the nested chinese restaurant process. In *NIPS*, 2004.

[24] Hastings. Monte Carlo sampling methods using markov chains and their applications. 1970.

[25] Hoffman et al. Content-based musical similarity computation using the hierarchical Dirichlet process. In *ISMIR*, 2008.

[26] Hughes et al. Reliable and scalable variational inference for the hierarchical Dirichlet process. In *AISTATS*, 2015.

[27] Jain and Neal. A split-merge Markov chain Monte Carlo procedure for the Dirichlet process mixture model. *JCGS*, 2004.

[28] Liang et al. The infinite pcfg using hierarchical dirichlet processes. In *EMNLP-CoNLL*, 2007.

[29] Ochs et al. Segmentation of moving objects by long term video analysis. *IEEE TPAMI*, 2013.

[30] Paisley et al. Nested hierarchical Dirichlet processes. *IEEE TPAMI*, 2015.

[31] Parisi and Perego. Flexible parallel split-merge MCMC for the HDP. In *ICBSA*, 2016.

[32] Pedregosa et al. Scikit-learn: Machine learning in Python. *JMLR*, 2011.

[33] Pitman et al. Combinatorial stochastic processes. Technical report, Technical Report 621, Dept. Statistics, UC Berkeley. Lecture notes, 2002.

[34] Ren et al. The dynamic hierarchical Dirichlet process. In *ICML*, 2008.

[35] S. Rogers, A. Klami, J. Sinkkonen, M. Girolami, and S. Kaski. Infinite factorization of multiple non-parametric views. *Machine Learning*, 2010.

[36] Smyth et al. Asynchronous distributed learning of topic models. In *NIPS*, 2009.

[37] E. B. Sudderth. *Graphical models for visual object recognition and tracking*. PhD thesis, MIT, 2006.

[38] Tayal et al. Hierarchical double Dirichlet process mixture of Gaussian processes. In *AAAI*, 2012.

[39] Teh et al. Sharing clusters among related groups: Hierarchical Dirichlet processes. In *NIPS*, 2005.

[40] Teh et al. Hierarchical Dirichlet processes. *JASA*, 2006.

[41] Teh et al. Collapsed variational inference for HDP. In *NIPS*, 2008.

[42] Uziel et al. Bayesian adaptive superpixel segmentation. In *ICCV*, 2019.

[43] Wang and Blei. A split-merge MCMC algorithm for the hierarchical Dirichlet process. *arXiv:1201.1657*, 2012.

[44] Wang and Blei. Truncation-free online variational inference for Bayesian nonparametric models. In *NIPS*, 2012.

[45] Wang et al. Online variational inference for the hierarchical Dirichlet process. In *AISTATS*, 2011.

[46] West and Escobar. *Hierarchical priors and mixture models, with application in regression and density estimation*. Duke University, 1993.

[47] Williamson et al. Parallel Markov chain Monte Carlo for nonparametric mixture models. In *ICML*, 2013.

[48] Xuan et al. Cooperative hierarchical Dirichlet processes: Superposition vs. maximization. *Artificial Intelligence*, 2019.