Hierarchical Bayesian Models for Unsupervised Scene Understanding

Daniel M. Steinberg*, Oscar Pizarro, Stefan B. Williams

Australian Centre for Field Robotics
The University of Sydney
Sydney, NSW 2006

Abstract

For very large datasets with more than a few classes, producing ground-truth data can represent a substantial, and potentially expensive, human effort. This is particularly evident when the datasets have been collected for a particular purpose, e.g. scientific inquiry, or by autonomous agents in novel and inaccessible environments. In these situations there is scope for the use of unsupervised approaches that can model collections of images and automatically summarise their content. To this end, we present novel hierarchical Bayesian models for image clustering, image segment clustering, and unsupervised scene understanding. The purpose of this investigation is to highlight and compare hierarchical structures for modelling context within images based on visual data alone. We also compare the unsupervised models with state-of-the-art supervised and weakly supervised models for image understanding. We show that some of the unsupervised models are competitive with the supervised and weakly supervised models on standard datasets. Finally, we demonstrate these unsupervised models working on a large dataset containing more than one hundred thousand images of the sea floor collected by a robot.

Keywords:
Scene understanding, unsupervised learning, clustering, hierarchical Bayesian models, topic models, variational Bayes

1. Introduction

In many real-world applications involving the collection of visual data, obtaining ground truth from a human expert can be very costly or even infeasible. For example, remote autonomous agents operating in novel environments like extra-planetary rovers and autonomous underwater vehicles (AUVs) are very effective at collecting huge quantities of visual data. Sending all of this data back to human operators quickly is hard since communication is usually bandwidth limited. In these situations it may be desirable to have algorithms operating on these vehicles that can summarise the data in unsupervised but semantically meaningful ways.

Similarly, many scientific datasets may contain terabytes of visual data that require expert knowledge to label it in a manner which is suitable for scientific inference. Obtaining such knowledge for large datasets can be a large drain on research resources. Again, it would be desirable to have algorithms that can separate this data automatically and in semantically meaningful ways, so the attention of the domain experts can be focused on subsets of the visual data for further labelling. In section 6 we present a large visual dataset collected by an AUV that exhibits exactly this problem.

Recently there has been much focus on the computer vision problem of *scene understanding*, whereby multiple sources of information and various contextual relationships are used to create holistic scene models. Typically the aim in scene understanding is to improve scene recognition tasks while taking advantage of scene labels or annotations [1–4], accompanying caption or body text [5], or even contextual relationships between image labels and low-level visual features [6, 7]. Most of these approaches are weakly supervised, semi-supervised or supervised in nature, and not much attention has been given to fully unsupervised, visual-data only holistic scene understanding.

In this article we wish to explore how unsupervised, or visual-data only, techniques can be applied to the problem of scene understanding. To this end we experiment with well established unsupervised models for clustering, such as Bayesian mixture models [8] and latent Dirichlet allocation [9]. These models cluster coarse whole-image descriptors, or cluster individual parts of images (but not simultaneously). We also explore models that can cluster data on multiple levels simultaneously (e.g. image segments or parts, and images), which are similar to the models presented in [4, 10]. These models discover the relationships between objects in images, and then define scene types as distributions of these objects. Also, by knowing the scene type, contextual information is used to aid in finding objects within scenes. Finally we present a new model that can cluster multiple sources of visual information, such as segment and image descriptors. This model takes advantage of holistic image descriptors, which may encode spatial layout, as well as modelling scene types as distributions of objects.

All of these models are compared on standard computer vi-

*Corresponding author
Email addresses: d.steinberg@acfr.usyd.edu.au (Daniel M. Steinberg), o.pizarro@acfr.usyd.edu.au (Oscar Pizarro), stefanw@acfr.usyd.edu.au (Stefan B. Williams)

sion datasets as well as a large AUV dataset for scene and object discovery. Emphasis is placed on scene category discovery, since we compare these unsupervised methods to state-of-the-art weakly-, semi- and supervised techniques for scene understanding. We also compare these models for object discovery in two of the experiments.

In the next section we review the most relevant literature to place this work in context. We then present the hierarchical Bayesian models we use for unsupervised scene understanding in section 3 and in section 4 we present variational Bayes algorithms for learning these models. In section 5 we describe the image and image-segment descriptors we use, since these play a large part in the performance of these purely visual-data driven models. Then in section 6 we empirically compare all of the aforementioned models, and summarise our results in section 7.

2. Relevant Literature

Visual context, such as the spatial structure of images, and position and co-occurrence of objects within scenes provide semantic information that aids object and scene recognition in our visual cortex [11, 12]. Similarly, semantic information about images can be derived from the volumes of textual data that accompanies these images in the form of tags, captions and paragraph text on the Internet. Consequently, there has recently been a lot of research focusing on holistic image "understanding", where these sources of information are fused in order to improve scene and object recognition tasks.

An early attempt at combining annotation information with scene modelling proposed in [1] extends latent Dirichlet allocation (LDA) [9] to use both visual and textual data for inferring image tags in untagged images. This is essentially a "weak" form of supervision, where the exact image classes are unknown, but some semantically relevant information is still used in training the model. Subsequent research, [2–4, 13–15] (amongst others) present hierarchical Bayesian models that can simultaneously classify scenes and recognise objects. These models can be supervised at the scene level, object level, or both. They can also use "weak" labels, or annotations, at the image or object levels [2–4]. Typically the features used to represent each image in these models are the proportions of superpixel clusters (objects) contained within the images. The superpixels are usually described by a combination of bag-of-words (BoW) features, such as quantised SIFT descriptors and quantised attributes like colour, texture and shape.

Li et al. [3] present a hierarchical Bayesian model that has a principled way of dealing with "noisy" or irrelevant object tags. Essentially a trade off is made between the model's certainty of the distribution of tags that correspond to a visual object class, and the distribution of tags that are irrelevant to the current object class. If an object class has a strong associated posterior distribution over the corresponding tags, a new tag that has low likelihood under this posterior is likely to be declared as irrelevant by an indicator variable. This model can also infer tags for images when they are missing. Quite a different approach to modelling textual and visual information is presented in [5]. They use a kernel canonical correlation analysis model (CCA) that attempts to learn the latent subspace that connects visual features with unaligned text (e.g. web pages with images). They then use this learned subspace for scene classification tasks.

Fei-Fei and Li [16] also present a model where the scene and object levels are classified in the same framework, but are linked through a higher "event" level, such as a particular sporting event. For example, the objects in an image may be a person, skis etc., and the scene may be of a snowy mountain. Naturally, these are both related to a "skiing" event, which is simultaneously inferred. These higher level contextual relationships were shown to aid image classification. Similarly, it has been found in works such as [17, 18] that knowledge of scene-type context can aid object recognition. In [17] the authors use a hidden Markov model (HMM) to classify a scene, and give certain objects a-priori more probability of being detected conditioned on the scene type. For example, it is more likely you would find a coffee machine in a kitchen. Similarly, certain objects commonly co-occur, and so detection of one object (street) may be used to aid detection of another object (building), as demonstrated by [19]. They use tree-like models to infer the contextual and spatial relationships of, and between, labelled objects to aid inference in unlabelled test sets. It is worth noting at this juncture that while object discovery can be an important part of scene understanding, emphasis has usually been placed on scene recognition in the scene understanding literature. Object recognition and discovery performance is usually presented in a qualitative fashion. Conversely, scene recognition is not given much attention in the object recognition and discovery literature, which focuses on quantitative measures of recognition and object purity.

Many models for scene understanding explicitly model the spatial layout of scenes [14, 15, 19, 20], or may make use of non-parametric processes or random fields to enforce segment-label contiguity [4, 21–24]. In [14] the authors present a supervised model, the context-aware topic model (CA-TM), that is similar to hierarchical Bayesian models like [13], but it also learns the absolute (as opposed to relative) position of objects within a scene type. For example, it learns that sky objects are at the top, and buildings are at the sides, of street scenes etc. Hence it takes advantage of both scene and spatial context for classification and object recognition. It can be both supervised at the scene and, optionally, at the object level. A model with a similar concept is presented in [20], however their emphasis is on object detection/discovery rather than scene recognition. The models presented in this article do not explicitly model scene spatial layout, however, the image descriptors used encode this information, see section 5 for details.

Recently [6] combined Beta-process sparse-code dictionary learning, topic modelling and image classification in one generative framework. Essentially this framework models images from the pixel level to scene level. This is quite an impressive feat, and results in a *very* complex model. This model can also be used for unsupervised image clustering, but not necessarily object detection/segmentation. It can also use image annotations where available. While the classification results are impressive, each iteration of learning (Gibbs sampling) takes on

the order of minutes, when it is usually milliseconds or seconds for other models. A similar concept is presented in [7], however they use a Bayesian co-clustering framework to incorporate semantic knowledge from image labels for visual dictionary learning. They can directly relate image features to semantic concepts, and show better performance than [6].

Scene understanding is a very active area of research, however much of the literature is concerned with weakly supervised, semi-supervised (a few strong labels) or supervised approaches to image understanding. Some of the aforementioned models can be used in a fully unsupervised, visual data only setting, though they may operate in a reduced capacity. For instance, the model in [13] loses its ability to perform scene recognition/discovery and reverts to just clustering segments when image labels are not present. Also [4] and [6] can be used as unsupervised models when no annotation data is available, however they were not rigorously tested in such situations. The only publication, to the authors' knowledge, that presents a model exclusively designed for unsupervised scene understanding is [25]. This model is also reviewed and used for comparison in this work.

There has been more work on unsupervised object discovery, where scene recognition/discovery is not an important consideration. For instance [26] and [20] cluster segments from multiple image segmentations in order to find the "purest" instances of objects. Though [20] can also make use of object spatial layout, and labelled categories where available, which is a similar approach taken by [27]. A comprehensive review of clustering models such as K-means and spectral clustering, and topic models such as LDA and non-negative matrix factorisation (NMF) applied to object discovery is provided in [28]. They test these models on single and multiple object per image tasks, and with different BoW feature normalisations. Similarly, there has been much work on unsupervised scene discovery, where typically whole-scene descriptors are used without explicitly modelling image parts [29–31].

From the aforementioned literature it is apparent that performance for scene and object recognition can be greatly increased by taking advantage of joint scene and object contextual cues. However, it is also apparent that not much attention has been given to achieving holistic scene understanding in a completely unsupervised manner. The work presented in this article reviews and introduces various approaches for a more fully-fledged unsupervised scene understanding framework in the absence of any annotations or related textual information.

3. Bayesian models for Unsupervised Scene Understanding

In this section we present and discuss the structure of a number of hierarchical Bayesian models of increasing complexity that we apply to unsupervised scene understanding tasks. We start with Bayesian Gaussian mixture models (BGMMs) [8, 32] and latent Dirichlet allocation [9], but with Gaussian clusters or topics (G-LDA), for scene or segment clustering. We then present two novel models for simultaneous image and segment clustering. The first is the simultaneous clustering model (SCM), which is similar to the models presented in [4] and [10]. The second is the multiple-source clustering model (MCM) that can cluster both image and segment descriptors.

3.1. Bayesian Gaussian Mixture Models

We will present BGMMs in the context of clustering images, but these models can equally be applied to clustering segments.

Firstly, a BGMM assumes all images in a dataset, $\mathbf{W} = \{\mathbf{w}_i\}_{i=1}^{I}$ where $\mathbf{w}_i \in \mathbb{R}^{D_{\text{im}}}$, are drawn from a weighted sum of T Gaussian distributions;

$$\mathbf{w}_i \sim \sum_{t=1}^{T} \pi_t \mathcal{N}\left(\mathbf{w}_i | \boldsymbol{\eta}_t, \boldsymbol{\Psi}_t^{-1}\right). \tag{1}$$

Here $\boldsymbol{\pi} = \{\pi_t\}_{t=1}^{T}$ are the mixture weights, where $\pi_t \in [0, 1]$ and $\sum_{t=1}^{T} \pi_t = 1$. Also, $\boldsymbol{\eta}_t$ and $\boldsymbol{\Psi}_t$ are the means and inverse covariances (precisions) for each Gaussian cluster respectively.

An auxiliary indicator variable is also introduced, $\mathbf{Y} = \{y_i\}_{i=1}^{I}$ where $y_i \in \{1, \ldots, T\}$, which assigns each observation to a Gaussian component according to the following conditional relationship;

$$\mathbf{w}_i | y_i \sim \prod_{t=1}^{T} \mathcal{N}\left(\mathbf{w}_i | \boldsymbol{\eta}_t, \boldsymbol{\Psi}_t^{-1}\right)^{\mathbf{1}[y_i = t]}. \tag{2}$$

Here $\mathbf{1}[\cdot]$ is an indicator function that evaluates to 1 if the expression in the brackets is true, or 0 otherwise. The y_i are distributed according to a Categorical distribution,

$$y_i \sim \text{Categ}(\boldsymbol{\pi}) = \prod_{t=1}^{T} \pi_t^{\mathbf{1}[y_i = t]}. \tag{3}$$

Because this is a *Bayesian* model, prior distributions are placed on all of the model parameters as well,

$$\boldsymbol{\pi} \sim \text{Dir}(\alpha), \tag{4}$$

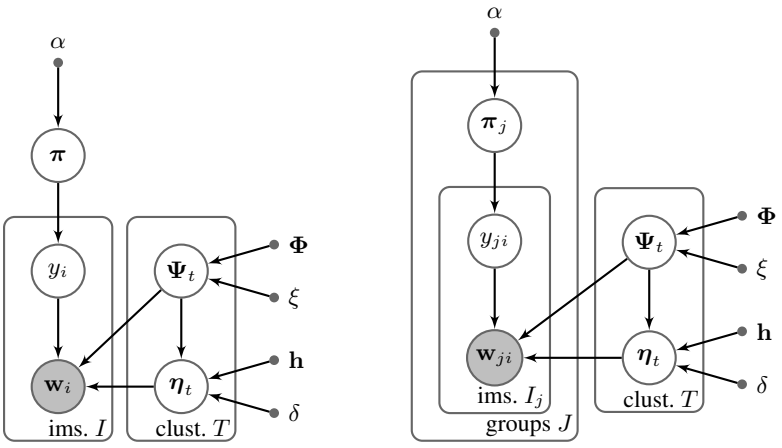
$$\boldsymbol{\eta}_t \sim \mathcal{N}\left(\mathbf{h}, (\delta \boldsymbol{\Psi}_t)^{-1}\right), \tag{5}$$

$$\boldsymbol{\Psi}_t \sim \mathcal{W}(\boldsymbol{\Phi}, \xi), \tag{6}$$

where $\mathcal{W}(\cdot)$ is a Wishart distribution, and only a single scalar parameter is given to the Dirichlet distribution as shorthand for a symmetric Dirichlet prior.

The following describes the generative process of the Bayesian Gaussian Mixture model:

1. Draw T cluster parameters $\boldsymbol{\eta}_t$ and $\boldsymbol{\Psi}_t$ from (5) and (6) respectively.
2. Draw mixture weights $\boldsymbol{\pi} \sim \text{Dir}(\alpha)$.
3. For each image, $i \in \{1, \ldots, I\}$:
   (a) Choose an image cluster $y_i \sim \text{Categ}(\boldsymbol{\pi})$.
   (b) Draw an observation from the chosen cluster $\mathbf{w}_i | (y_i = t) \sim \mathcal{N}(\boldsymbol{\eta}_t, \boldsymbol{\Psi}_t)$.



(a) Bayesian GMM (b) Gaussian LDA

Figure 1: Graphical models of (a) a Bayesian Gaussian mixture model (BGMM), and (b) Gaussian latent Dirichlet allocation (G-LDA). Circles denote random variables, shaded circles are observable random variables. The plates denote replication of encompassed entities, and the points represent point estimates of the model hyper-parameters.

The graphical model of this process is shown in Figure 1. The actual (posterior) hyper-parameters ($\tilde{\alpha}_t$, $\tilde{\mathbf{h}}_t$, etc.), number of clusters (T), and indicator assignments (y_i) are learned using variational Bayes [8], which is discussed in section 4.

There exist generalisations of this BGMM where $T \to \infty$ using a Dirichlet process instead of a symmetric Dirichlet over the mixture weights [33]. Such a model is the variational Dirichlet Process (VDP) presented in [32]. We will not describe such a model here, however we do use the VDP in the experiments in section 6. We have found that the variational Bayes realisation of the VDP and BGMM yield very similar results.

The BGMM and VDP do not take advantage of any contextual or structural information when applied to clustering images or segments. They simply cluster images or segments as if they were all in one "bag". We use the BGMM/VDP as baseline Bayesian unsupervised methods for comparison to other, more sophisticated methods.

3.2. Gaussian Latent Dirichlet Allocation

Latent Dirichlet allocation (LDA) [9] was originally formulated for modelling text, and typically has a Multinomial or Categorical cluster (topic) distribution. Because the image and segment descriptors we have used in this article are best modelled with Gaussian clusters, we now present a version of smoothed LDA with Gaussian clusters (G-LDA).

Using again the application of clustering images, G-LDA assumes data originates in J distinct groups or photo albums, $\mathbf{W} = \{\mathbf{W}_j\}_{j=1}^{J}$ (these groups are known as "documents" in the text modelling literature). Each of these albums contains I_j images, $\mathbf{W}_j = \{\mathbf{w}_{ji}\}_{i=1}^{I_j}$. Also, the mixture weights are specific to each group, $\{\boldsymbol{\pi}_j\}_{j=1}^{J}$, but the clusters are shared between all groups,

$$\mathbf{w}_{ji} \sim \sum_{t=1}^{T} \pi_{jt} \mathcal{N}\left(\mathbf{w}_{ji} | \boldsymbol{\eta}_t, \boldsymbol{\Psi}_t^{-1}\right). \tag{7}$$

The differences between G-LDA and the BGMM are perhaps illustrated more clearly in Figure 1, and by the following generative process:

1. Draw T cluster parameters $\boldsymbol{\eta}_t$ and $\boldsymbol{\Psi}_t$ from (5) and (6) respectively.
2. For each group or album, $j \in \{1, \ldots, J\}$:
   (a) Draw mixture weights $\boldsymbol{\pi}_j \sim \mathrm{Dir}(\alpha)$.
   (b) For each image, $i \in \{1, \ldots, I_j\}$:
      i. Choose an image cluster $y_{ji} \sim \mathrm{Categ}(\boldsymbol{\pi}_j)$.
      ii. Draw an observation from the chosen cluster $\mathbf{w}_{ji} | (y_{ji} = t) \sim \mathcal{N}(\boldsymbol{\eta}_t, \boldsymbol{\Psi}_t)$.

G-LDA is quite similar to the BGMM except that is has mixture weights specific to each group or album. So each album of images has a specific proportion of scene-types (image clusters). Alternatively, if we used this model to cluster image segments, then each image would be described as a particular proportion of objects (segment-clusters). Hence, G-LDA models album context for image clustering, and image context when applied to segment clustering.

Generalisations of LDA to $T \to \infty$ also exist, such as the hierarchical Dirichlet process (HDP) [34]. These models typically aid in the selection of T because of the hierarchical nature of the prior used. However, we have found this not to be an issue with G-LDA because of the heavy complexity penalties introduced by the Gaussian cluster priors (see section 4 and Appendix A). Hence, we have elected to stay with the more simple, conjugate LDA-based model for this article.

Most standard computer vision datasets are not divided into photo albums, and so in section 6 we mostly use G-LDA for clustering segments. However, the AUV dataset is comprised of multiple surveys, which we do use as albums.

3.3. Simultaneous Clustering Model

Now we present the novel simultaneous clustering model (SCM), which can simultaneously cluster image segments *and* images unlike the BGMM and G-LDA. Like G-LDA, the SCM models albums j, but does not explicitly model image descriptors ($\mathbf{w}_{ji}$). Instead it models images as distributions of objects, or segment-clusters. This is a "bag-of-segments" representation since the layout, or order, of the segments in the image is not modelled.

Each image is comprised of N_{ji} non-overlapping segments, $\mathbf{X}_{ji} = \{\mathbf{x}_{jin}\}_{n=1}^{N_{ji}}$ where $\mathbf{x}_{jin} \in \mathbb{R}^{D_{\text{seg}}}$, which are drawn from a mixture of K Gaussians or "object" types. The segment cluster weights $\boldsymbol{\beta}_t = \{\beta_{t1}, \ldots, \beta_{tK}\}$ are *specific to each scene-type*, t, as opposed to each image in the case of G-LDA;

$$\text{SCM}: \quad \mathbf{x}_{jin} | (y_{ji} = t) \sim \sum_{k=1}^{K} \beta_{tk} \mathcal{N}\left(\mathbf{x}_{jin} | \boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k^{-1}\right), \quad (8)$$

$$\text{G-LDA}: \quad \mathbf{x}_{jin} \sim \sum_{k=1}^{K} \beta_{ik} \mathcal{N}\left(\mathbf{x}_{jin} | \boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k^{-1}\right). \quad (9)$$

So where G-LDA models segments as a Gaussian mixture specific to each image, i, the SCM models segments as a Gaussian mixture specific to a scene type or cluster of images, t. Like in the BGMM, an indicator variable for each segment observation, $z_{jin} \in \{1, \ldots, K\}$, is used to assign the observation to a segment-cluster (object-type). This indicator variable also has a Categorical distribution, but is conditioned on the scene indicator,

$$z_{jin} | y_{ji} \sim \prod_{t=1}^{T} \text{Categ}(z_{jin} | \boldsymbol{\beta}_t)^{\mathbb{1}[y_{ji}=t]}, \quad (10)$$

note how this is similar to (2). Consequently, each image is described as a set of object types, $\mathbf{Z}_{ji} = \{z_{jin}\}_{n=1}^{N_{ji}}$, which is inherently a Multinomial distribution. This means that each scene-type, t, will have its own unique distribution of objects, $\boldsymbol{\beta}_t$. The BGMM and G-LDA represent a scene-type as a Gaussian cluster, whereas the SCM represents a scene-type as a Multinomial cluster.

All the SCM parameters have prior distributions;

$$\boldsymbol{\pi}_j \sim \text{GDir}(a, b), \quad (11)$$

$$\boldsymbol{\beta}_t \sim \text{Dir}(\theta), \quad (12)$$

$$\boldsymbol{\mu}_k \sim \mathcal{N}\left(\mathbf{m}, (\gamma \boldsymbol{\Lambda}_k)^{-1}\right), \quad (13)$$

$$\boldsymbol{\Lambda}_k \sim \mathcal{W}(\boldsymbol{\Omega}, \rho). \quad (14)$$

We have chosen to use a generalised Dirichlet distribution, $\text{GDir}(\cdot)$, [35, 36] over the scene-type mixture weights. It can be represented as a truncated stick breaking process, which is also used to approximate a Dirichlet process [37],

$$\pi_{jt} = v_{jt} \prod_{s=1}^{t-1} (1 - v_{js}), \quad v_{jt} \sim \begin{cases} \text{Beta}(a, b) & \text{if } t < T \\ 1 & \text{if } t = T. \end{cases} \quad (15)$$

where $v_{jt} \in [0, 1]$ are "stick-lengths" for each album. Here we have also elected to just choose a scalar value for the hyper-parameters a and b, like in the case of the symmetric Dirichlet. We use a generalised Dirichlet for the SCM because it is a heavier prior (has twice the number of parameters) than the symmetric Dirichlet. This helps variational Bayes select an appropriate number of scene types, T, when using a Multinomial scene-type representation. This is similar in concept to using a HDP, but without the complexity. This modelling choice is explored empirically in section 6.

The entire generative model of the SCM is represented in Figure 2, and is given below;

1. Draw T image cluster parameters $\boldsymbol{\beta}_t$ from (12).
2. Draw K segment cluster parameters $\boldsymbol{\mu}_k$ and $\boldsymbol{\Lambda}_k$ from (13) and (14) respectively.
3. For each group or album, $j \in \{1, \ldots, J\}$:
   (a) Draw mixture weights $\boldsymbol{\pi}_j \sim \text{GDir}(a, b)$.
   (b) For each image, $i \in \{1, \ldots, I_j\}$:
      i. Choose an image cluster $y_{ji} \sim \text{Categ}(\boldsymbol{\pi}_j)$.
      ii. For each image segment $n \in \{1, \ldots, N_{ji}\}$:
         A. Choose a segment cluster $z_{jin} | (y_{ji} = t) \sim \text{Categ}(\boldsymbol{\beta}_t)$.
         B. Draw an observation from the segment cluster $\mathbf{x}_{jin} | (z_{jin} = k) \sim \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k)$.

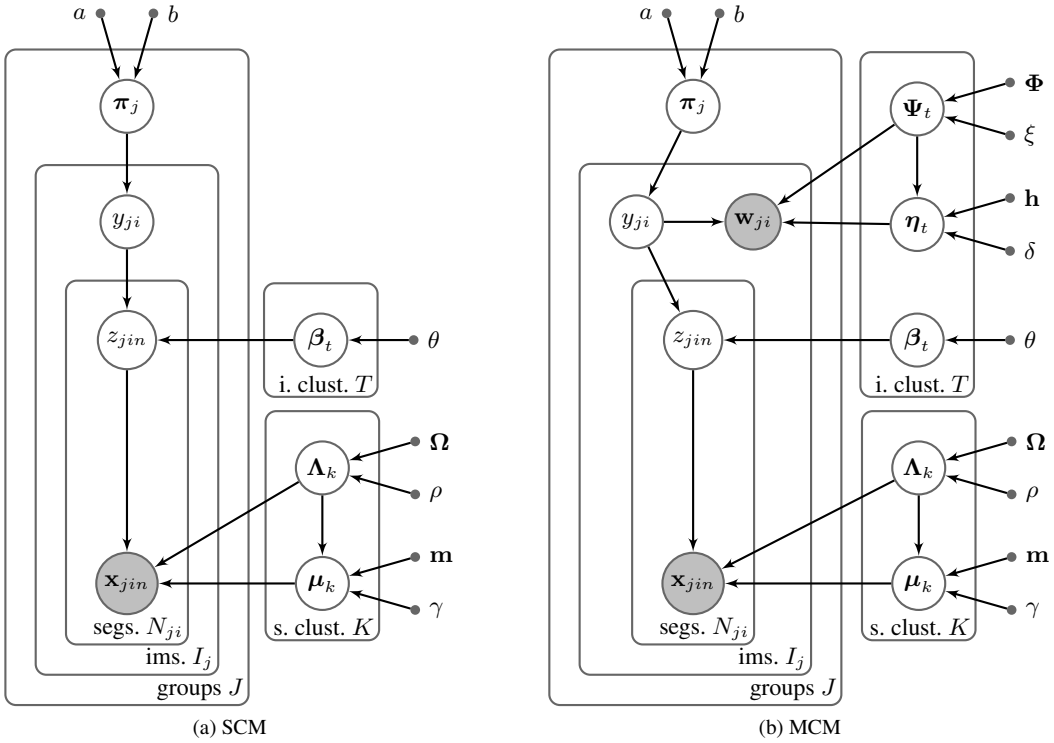
The SCM is similar in some ways to the model presented in [4], for instance it represents an image as a distribution of object types. However, the SCM retains the image-album context of G-LDA (when the latter is applied to clustering images). The SCM models segments as having scene-type context, i.e. a tree is more likely to appear in a forest scene than an indoor scene. This is unlike G-LDA (when applied to segments), which only models segments as having specific image context. Hence the SCM has better object co-occurrence modelling facility than G-LDA. Distributions of objects are captured at the scene-type level, $\boldsymbol{\beta}_t$, which involves many images, as opposed to just the single image level, $\boldsymbol{\beta}_i$.

3.4. Multiple-source Clustering Model

The final model we present in this article is the multiple-source clustering model (MCM). This model has also been presented in [25]. It essentially combines the SCM segment representation with the image-level representation of G-LDA.

Like the SCM, the MCM models segments, $\mathbf{x}_{jin}$, as a scene-type specific mixture of Gaussians, $\boldsymbol{\beta}_t$. But unlike the SCM, the MCM also models image descriptors, $\mathbf{w}_{ji}$, as a group specific mixture of Gaussians, like image-level G-LDA. So now scene clusters, or types, are represented by *both* the proportions of objects (segment clusters) within them, $\boldsymbol{\beta}_t$, and a Gaussian component that describes the overall scene appearance, parameterised by $\boldsymbol{\eta}_t$ and $\boldsymbol{\Psi}_t$.

The difference between the SCM and the MCM can be visualised by the graphical models in Figure 2. We also illustrate



(a) SCM (b) MCM

Figure 2: Graphical models of (a) the simultateous clustering model (SCM), and (b) the multiple-source clustering model (MCM).

how images are modelled under the SCM and MCM in Figure 3. The generative process of the MCM is;

1. Draw T image cluster parameters $\boldsymbol{\beta}_t$, $\boldsymbol{\eta}_t$ and $\boldsymbol{\Psi}_t$ from (12), (5) and (6) respectively
2. Draw K segment cluster parameters $\boldsymbol{\mu}_k$ and $\boldsymbol{\Lambda}_k$ from (13) and (14) respectively.
3. For each group or album, $j \in \{1, \ldots, J\}$:
   (a) Draw mixture weights $\boldsymbol{\pi}_j \sim \text{GDir}(a, b)$.
   (b) For each image, $i \in \{1, \ldots, I_j\}$:
      i. Choose an image cluster $y_{ji} \sim \text{Categ}(\boldsymbol{\pi}_j)$.
      ii. Draw an image observation from the chosen image cluster $\mathbf{w}_{ji} | (y_{ji} = t) \sim \mathcal{N}(\boldsymbol{\eta}_t, \boldsymbol{\Psi}_t)$.
      iii. For each image segment $n \in \{1, \ldots, N_{ji}\}$:
         A. Choose a segment cluster $z_{jin} | (y_{ji} = t) \sim \text{Categ}(\boldsymbol{\beta}_t)$.
         B. Draw a segment observation from the segment cluster $\mathbf{x}_{jin} | (z_{jin} = k) \sim \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k)$.

The type of context that the MCM models is similar to the SCM. The only real difference being that scene types have a joint Multinomial-Gaussian representation. This allows the MCM to more effectively model global scene attributes that may not be captured by just object co-occurrence. For instance, the image descriptors introduced in section 5 capture the coarse spatial structure of an image.

4. Variational Inference for the SCM and MCM

In this section variational Bayes inference algorithms are derived for learning the posterior latent variables of the SCM and MCM, i.e., posterior hyper-parameters, labels, and number of clusters. We do not present the variational Bayes algorithms for the BGMM, VDP or G-LDA since these can be found in [8, 32, 38, 39]. Also, many of the updates are similar between these models.

Typically, to learn the posterior latent variables of the types of Bayesian models presented in the previous section, the models' log-marginal likelihood is maximised with respect to the set of model hyper-parameters, $\boldsymbol{\Xi}$. The log-marginal likelihood takes the general form:

$$\log p(\mathbf{X}|\boldsymbol{\Xi}) = \log \int p(\mathbf{X}, \mathbf{Z}, \boldsymbol{\Theta}|\boldsymbol{\Xi}) \, d\mathbf{Z} \, d\boldsymbol{\Theta}, \qquad (16)$$

where $\mathbf{X}$ are observable, $\mathbf{Z}$ are latent indicators, and $\boldsymbol{\Theta}$ are the set of model parameters. This integral is intractable in the case of all of the presented models, and so an approximation of this marginal log-likelihood is usually made. In the case of variational Bayes, this approximation is called *free energy*, $\mathcal{F}$. The approximation starts by representing the posterior distribution of the model latent variables with a set of factored distributions:

$$p(\mathbf{Z}, \boldsymbol{\Theta}|\mathbf{X}, \boldsymbol{\Xi}) \approx q(\mathbf{Z}) \, q(\boldsymbol{\Theta}) \, . \qquad (17)$$

By optimising the free energy functional $\mathcal{F}[q(\mathbf{Z}) \, , q(\boldsymbol{\Theta})]$, the Kullback-Leibler divergence, $\text{KL}[p\|q]$, is minimised between the approximation and the true posterior. This optimisation

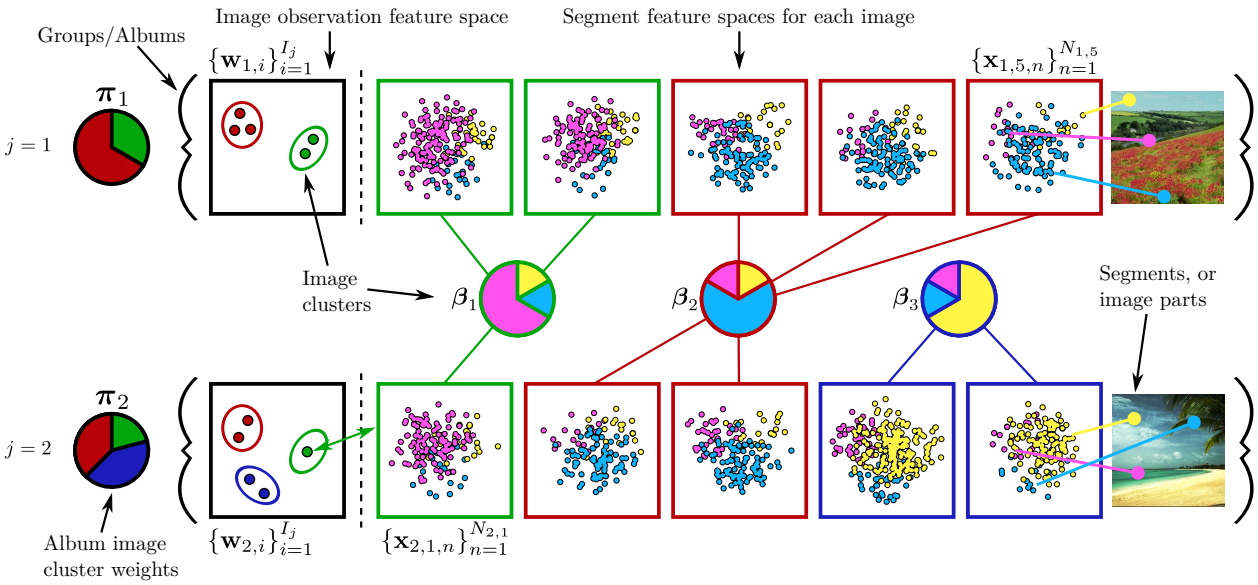


Figure 3: Demonstration of how the MCM clusters multiple observation sources in groups. The coloured points (cyan, yellow, magenta) in each coloured square represent an image's segment descriptors, $\mathbf{x}_{jin}$, in segment descriptor feature space. Segment descriptors are clustered into object-types (cyan $\approx$ plant, magenta $\approx$ water and yellow $\approx$ sky), and are shared between images and groups. Similar images will have a similar distribution of these object-types, $\boldsymbol{\beta}_t$. These proportions are denoted by the pie charts with the coloured borders. The coloured points (red, green, blue) in the black squares represent the image descriptors, $\mathbf{w}_{ji}$, in image descriptor feature space. Each black square represents all of the images in a group/album. The coloured squares correspond to the red, green and blue points in the black squares, symbolising that both an image descriptor *and* proportions of object-types within an image describe each image. Groups can likewise be described by the proportions of image clusters within them, $\boldsymbol{\pi}_j$. The SCM is the same as this depiction, but does *not* make use of $\mathbf{w}_{ji}$.

typically results in an expectation maximisation-like algorithm [40] that alternates between finding the expected distribution (or assignments) for the indicators, $q(\mathbf{Z})$, and the optimum value of the variational posterior hyper-parameters, $\tilde{\boldsymbol{\Xi}}$, that govern $q(\boldsymbol{\Theta})$. The expectation and maximisation steps are alternated until $\mathcal{F}$ converges. For more information on this general variational Bayes algorithm see [38, 40]. Also, for the exact form of $\mathcal{F}$ for the SCM and MCM see Appendix A.

4.1. Simultaneous Clustering Model

Applying the variational Bayes learning algorithm to the SCM yields the following expectation step for the segment indicators, $\mathbf{Z}$,

$$q(z_{jin}=k) = \frac{1}{\mathcal{Z}_{z_{jin}}} \exp\Bigg\{ \sum_{t=1}^{T} q(y_{ij}=t)\, \mathbb{E}_{q_\beta}[\log \beta_{tk}] + \mathbb{E}_{q_{\boldsymbol{\mu},\boldsymbol{\Lambda}}}\left[\log \mathcal{N}\left(\mathbf{x}_{jin} | \boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k^{-1}\right)\right] \Bigg\}. \quad (18)$$

Here $\mathcal{Z}_{z_{jin}}$ is a normalisation constant which is straightforwardly computed from the sum over k of the un-normalised components of (18). Also, $\mathbb{E}_q[\cdot]$ denotes the expectation with respect to the variational distribution, $q(\cdot)$. These expectations are given in Appendix B. In (18) the term with the sum over the weighted image label probabilities, $q(y_{jin}=t)$, assigns more or less likelihood of the current segment observation belonging to the segment cluster based on the probability of the image belonging to a scene-type, t. For example, if the image is of a forest type, then the current object is more likely to be a tree trunk than a building. This is how scene-type context is modelled for objects.

Similarly, the following expectation step is obtained for the image indicators, $\mathbf{Y}$,

$$q(y_{ji}=t) = \frac{1}{\mathcal{Z}_{y_{ji}}} \exp\Bigg\{ \mathbb{E}_{q_\pi}[\log \pi_{jt}] + \sum_{k=1}^{K} \mathbb{E}_{q_\beta}[\log \beta_{tk}] \sum_{n=1}^{N_{ji}} q(z_{jin}=k) \Bigg\}. \quad (19)$$

Again, $\mathcal{Z}_{y_{ji}}$ is simply calculated by summing the un-normalised components over t, and the expectations are given in Appendix B. From (19) we can see that an image is assigned to a scene-type, t, by the number and co-occurrence of object types within it. This is indicated by the term containing sums over $q(z_{jin}=k)$, which is essentially a Multinomial log-likelihood.

Optimising $\mathcal{F}$ for the parameters of the SCM leads to the following variational posterior updates for the mixture weights, $\boldsymbol{\pi}_j$, and image cluster parameters, $\boldsymbol{\beta}_t$;

$$\begin{aligned} \tilde{a}_{jt} &= a + \sum_{i=1}^{I_j} q(y_{ji}=t)\,, \\ \tilde{b}_{jt} &= b + \sum_{i=1}^{I_j} \sum_{s=t+1}^{T} q(y_{ji}=s)\,, \\ \tilde{\theta}_{tk} &= \theta + \sum_{j=1}^{J} \sum_{i=1}^{I_j} q(y_{ji}=t) \sum_{n=1}^{N_{ji}} q(z_{jin}=k)\,. \end{aligned} \quad (20)$$

These updates are essentially just the prior with added observation counts, or sufficient statistics. The sum for $\tilde{b}_{jt}$ in (20)

must be performed in descending cluster size order, as per [32]. The variational posterior Gaussian-Wishart hyper-parameters for the segment clusters are,

$$\begin{aligned}
\tilde{\gamma}_k &= \gamma + N_k, \\
\tilde{\mathbf{m}}_k &= \frac{1}{\tilde{\gamma}_k}\left(\gamma\mathbf{m} + N_k\bar{\mathbf{x}}_k\right), \\
\tilde{\rho}_k &= \rho + N_k, \\
\tilde{\mathbf{\Omega}}_k^{-1} &= \mathbf{\Omega}^{-1} + N_k\mathbf{R}_k + \frac{\gamma N_k}{\tilde{\gamma}_k}\left(\bar{\mathbf{x}}_k - \mathbf{m}\right)\left(\bar{\mathbf{x}}_k - \mathbf{m}\right)^\top,
\end{aligned} \tag{21}$$

where,

$$\begin{aligned}
N_k &= \sum_{j=1}^{J}\sum_{i=1}^{I_j}\sum_{n=1}^{N_{ji}} q(z_{jin} = k), \\
\bar{\mathbf{x}}_k &= \frac{1}{N_k}\sum_{j=1}^{J}\sum_{i=1}^{I_j}\sum_{n=1}^{N_{ji}} q(z_{jin} = k)\,\mathbf{x}_{jin}, \\
\mathbf{R}_k &= \frac{1}{N_k}\sum_{j=1}^{J}\sum_{i=1}^{I_j}\sum_{n=1}^{N_{ji}} q(z_{jin} = k)\left(\mathbf{x}_{jin} - \bar{\mathbf{x}}_k\right)\left(\mathbf{x}_{jin} - \bar{\mathbf{x}}_k\right)^\top.
\end{aligned} \tag{22}$$

The expectation steps (18) and (19) and the maximization steps (21) are alternated until $\mathcal{F}$ for the SCM converges. $\mathcal{F}$ is given in Appendix A.

4.2. Multiple-source Clustering Model

Since the MCM is similar to the SCM, apart from the image observation model, many of the variational updates are the same. The image indicator, $\mathbf{Y}$, updates have a different form because of the image observations,

$$\begin{aligned}
q(y_{ji} = t) = \frac{1}{\mathcal{Z}_{y_{ji}}}\exp\Bigg\{&\mathbb{E}_{q_\pi}[\log\pi_{jt}] \\
&+ \sum_{k=1}^{K}\mathbb{E}_{q_\beta}[\log\beta_{tk}]\sum_{n=1}^{N_{ji}} q(z_{jin} = k) \\
&+ \mathbb{E}_{q_{\boldsymbol{\eta},\mathbf{\Psi}}}\left[\log\mathcal{N}\left(\mathbf{w}_{ji}|\boldsymbol{\eta}_t, \mathbf{\Psi}_t^{-1}\right)\right]\Bigg\}.
\end{aligned} \tag{23}$$

This equation has the same general form as (19), but with an added Gaussian log-likelihood term corresponding to the image descriptors, $\mathbf{w}_{ji}$.

The maximisation steps for the model parameters are also the same as the MCM apart from those for the Gaussian-Wishart prior over the image observation, $\mathbf{w}_{ji}$, clusters. However, these update equations are the same as those in (21), though the sums in (22) are only over j and i, and involve the image indicators, y_{ji}.

4.3. Split-tally Model Selection Heuristic

If the number of clusters, T and K, is known a-priori or set to some large value, the label and posterior hyper-parameter updates can simply be iterated until $\mathcal{F}$ converges to a local maximum. Some of the clusters will not accrue any observations because of the variational Bayes complexity penalties that naturally arise in $\mathcal{F}$. We have found that better clustering results can be obtained if we guide the search for the segment clusters.

The segment-cluster search heuristic we use is a much faster, greedy version of the exhaustive heuristic presented in [32]. The SCM and MCM start with $K = 1$ segment cluster, and iterate until convergence. Then the segment cluster is split in a direction perpendicular to its principal axis. The two resulting clusters are then refined by running variational Bayes over them for a limited number of iterations (we use a maximum of 15). $\mathcal{F}$ is estimated with this newly proposed split, and if it has increased in value, the split is accepted and the whole model is again iterated until convergence. Otherwise, the algorithm terminates. The exhaustive heuristic proceeds by trialling every possible cluster split between each model convergence stage, and only accepts the split that maximises $\mathcal{F}$. When K becomes large, this search heuristic becomes the dominant computational cost of the whole inference algorithm.

In our greedy "split-tally" heuristic, we guess which cluster to split first by ranking all clusters' approximate contribution to $\mathcal{F}$ (details in Appendix C). Also, a tally is kept of how many times a cluster has previously failed a split trial. Clusters that have not yet failed splits are prioritised for splitting. The first cluster split to increase $\mathcal{F}$ is accepted, and the tally for the original cluster is reset. All clusters must eventually fail to be split for the algorithm to terminate. We have found this split-tally heuristic greatly reduces run-time, without significantly impacting performance, mostly because of the tally. To our knowledge, this is the first time a tally has been used in such a heuristic. A similar heuristic was also trialled to search for T in the MCM, however we found that it was better to randomly initialise it to some large value, $T_{trunc} > T$, since both heuristics would interact. Also, there is no intuitive way to split the purely Multinomial image clusters of the SCM, so it is also randomly initialised.

We also use this split-tally heuristic for searching for the number of clusters in the VDP and G-LDA, when applied to image or segment clustering, in section 6.

4.4. Model Priors

Because all of the aforementioned models are Bayesian we need to choose priors in the form of initial hyper-parameter values. These priors are then updated as evidence is presented to the learning algorithm. By prioritising simplicity, we have chosen the following values for the prior hyper-parameters;

$$\begin{aligned}
\alpha, a, b, \theta, \gamma, \delta &= 1, \\
\rho &= D_{\text{seg}}, \\
\xi &= D_{\text{im}}, \\
\mathbf{m} &= \text{mean}(\mathbf{X}), \\
\mathbf{h} &= \text{mean}(\mathbf{W}), \\
\mathbf{\Omega} &= (\rho C_{w,s})^{-1}\mathbf{I}_{D_{\text{seg}}}, \\
\mathbf{\Phi} &= (\xi C_{w,i}\lambda^{\max}_{\text{cov}(\mathbf{W})})^{-1}\mathbf{I}_{D_{\text{im}}}.
\end{aligned} \tag{24}$$

The values for the first three equations in (24) have been chosen to be their minimum integer value in the support of their respec-

tive distributions. We have found that apart from θ, changing these values only has a minor affect on the posterior clusters. For the SCM in section 6 we do vary the value of θ.

The prior parameter, θ, essentially controls how many different objects (segment clusters) we expect to be in a particular scene-type a-priori. For low values of θ, we would expect only a few objects within each scene-type, i.e. we expect z_{jin} to only take a few values of k for a particular scene-type, t. Therefore more image clusters may be required to represent all possible object-types, k, since only a few can exist in a scene-type. We would expect the opposite to occur for high values of θ.

The Wishart matrix priors in the last two equations of (24) are just scaled identity matrices. This has the effect of making the algorithms expect isotropic Gaussian clusters in the data a-priori. Also $\lambda^{\max}_{\text{cov}(\mathbf{W})}$ is the largest Eigenvalue of the covariance of the image descriptors. This value is not used for the segment descriptors since they are whitened, see section 5. $C_{w,i}$ and $C_{w,s}$ (i for image, s for segment) are tunable parameters that encode the a-priori "width" of the isotropic clusters. These tuning parameters were found to have the largest effect on the number of (Gaussian) clusters found by the algorithms. In section 6 we will show clustering performance with varying θ, $C_{w,i}$ and $C_{w,s}$.

5. Image Representation

The aforementioned algorithms rely on highly discriminative visual descriptors since they are driven by visual-data alone. We have chosen unsupervised feature learning algorithms for this task as they are easily implemented and have excellent performance in a number of scene recognition tasks, e.g. [41].

5.1. Image Descriptors

We use a modified sparse coding spatial pyramid matching (ScSPM) [41] method to encode the image descriptors $\mathbf{w}_{ji}$. Figure 4 (a) demonstrates how these image descriptors are created. For all experiments we use the original 1024-base Caltech-101 dictionary supplied by [41] to encode dense 16×16 pixel SIFT patches with a stride of 8 pixels. We have found little to no reduction in classification and clustering performance using this pre-learned dictionary as compared to learning dictionaries for each specific dataset. This is similar to the observation made in [42].

We use orthogonal matching pursuit (OMP) with 10 activations in place of the original sparse coding encoding method used in [41] for the larger AUV dataset. It is much less computationally demanding and does not affect scene clustering performance greatly. We use the original pyramid with a [1,2,4] pooling region configuration, which leads to a 21,504 dimensional (sparse) code for each image. This is far too large to use with a Gaussian cluster model, but we have found that these codes are highly compressible with (randomised) PCA. Typically we can compress them to $D_{\text{im}} = 20$ while still achieving excellent image clustering performance.

5.2. Segment Descriptors

Out of the many segment descriptors tried, it was found that pooling dense independent component analysis (ICA) [43] codes within segments gave the best results. The following procedure was used to create a descriptor for each segment within an image:

1. Extract square patches centred on every pixel in the image.
2. (Optional) remove the DC offset, and contrast normalise the patches.
3. Use a random subset of all of the patches in the dataset to train an ICA dictionary, $\mathbf{D}$, and its pseudo-inverse, $\mathbf{D}^{+}$.
4. Use $\mathbf{D}^{+}$ to create a code (or filter response), $\mathbf{a}_l$, for all of the patches. This is a fast matrix multiplication operation, so is feasible for patches centred on every pixel, $l \in [1, L]$, in an image. L is the total number of pixels in an image.
5. Over-segment the image, obtaining sets of pixels S_{jin}. The results presented here used a fast mean-shift segmentation method [44].
6. Obtain segment descriptors by mean pooling all of the ICA dictionary responses in a segment in the following manner:

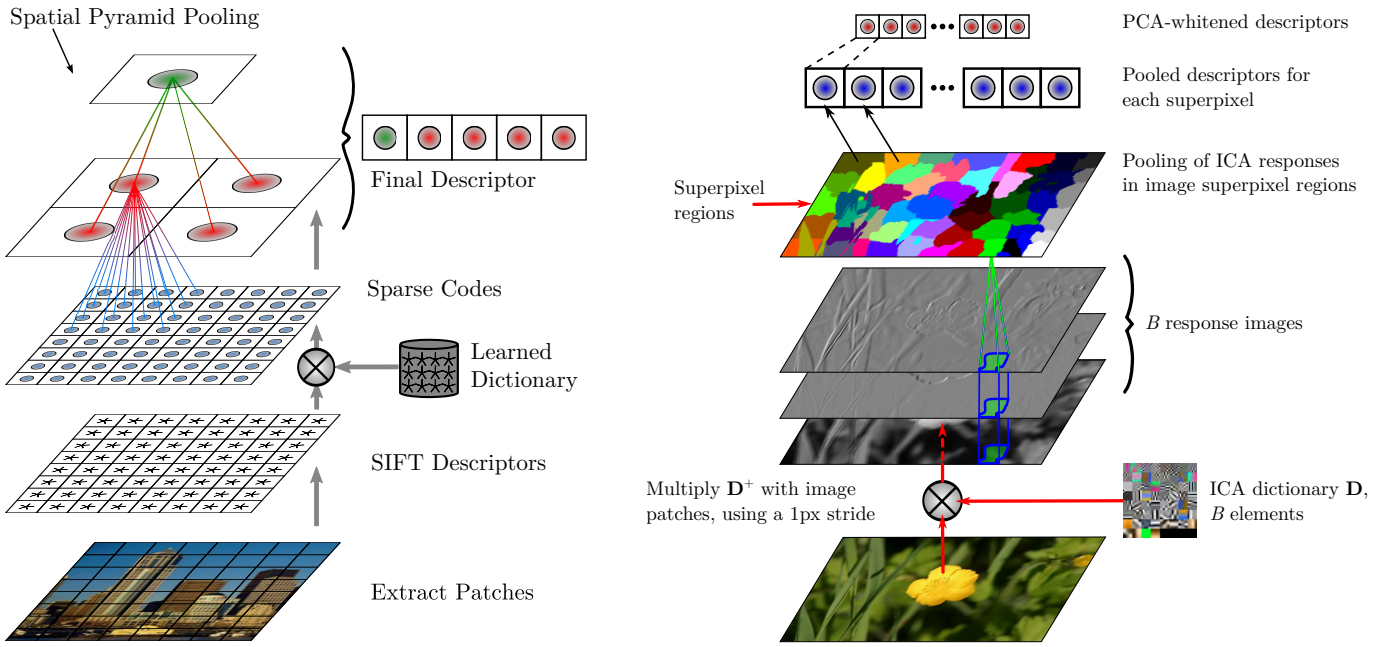
$$\tilde{\mathbf{x}}_{jin} = \frac{1}{\#S_{jin}} \sum_{l \in S_{jin}} \log |\mathbf{a}_l| \tag{25}$$

   These transformations greatly improved segment clustering performance. We conjecture that the absolute value makes the descriptors invariant to 90 degree phase shifts in $\mathbf{r}_l$. The logarithm transforms the range back to $(-\infty, \infty)$.
7. Obtain the final segment descriptors, $\mathbf{x}_{jin}$, by PCA whitening all the $\tilde{\mathbf{x}}_{jin}$. We perform dimensionality reduction as part of this whitening stage, to $D_{\text{seg}} = 15$, which preserves more than 90% of the spectral power.

This process is graphically demonstrated in Figure 4 (b).

A bag-of-words representation was not chosen for the segments as it would require a Multinomial cluster distribution as opposed to Gaussian. We found this representation to be less powerful for model selection in this unsupervised application (this is demonstrated in section 6 at the image level). We have chosen to leave a comprehensive comparison between these ICA-based features and bag-of-words features for object detection as future work. However, we do compare the image ScSPM representation against a bag-of-words image representation for use in spectral clustering in Table 1, where we can observe tangible benefits. Naturally, the performance of these algorithms is largely influenced by the representation chosen.

Both the image and segment descriptors take about 1 second each per image to calculate. The ScSPM and ICA features are complementary; the ScSPM descriptors encode the spatial layout and structural information of an image (the "gist"), whereas the ICA features encode fine-grained colour and texture information.



(a) ScSPM image descriptors

(b) ICA segment descriptors

Figure 4: (a) The sparse code spatial pyramid matching [41] image descriptors, and (b) the independent component analysis based segment descriptors. See text for details.

6. Experiments

In this section we compare the VDP, G-LDA, SCM and MCM in image and segment clustering tasks. We also compare to other unsupervised, weakly-supervised, semi-supervised and supervised algorithms in the literature for scene understanding/recognition. For this comparison we use three standard datasets (single album) and a large novel dataset consisting of twelve surveys (albums) from an autonomous underwater vehicle (AUV). We also explore whether a symmetric Dirichlet prior over group weights, $\boldsymbol{\pi}_j$, has an effect on clustering results for the SCM and MCM. Finally we explore whether modelling groups improves clustering performance on the AUV dataset.

Normalised mutual information (NMI) [45] is used to compare the clustering results to the ground truth image and segment labels. This is a fairly common measure in the clustering literature as it permits performance to be compared in situations where the number of ground truth classes and clusters are different. All results cited have been transformed into NMI scores from the confusion matrices given in their corresponding papers. This conversion is straight forward as long as the number of images used for testing within each class is known.

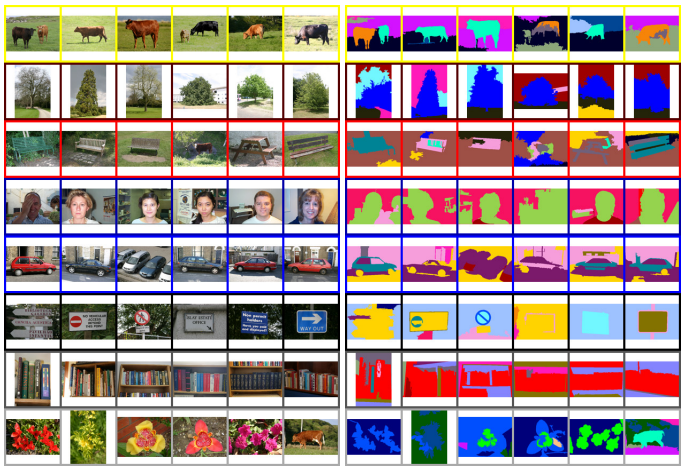
We also estimate the mean accuracy for the clustering results when benchmarking against supervised algorithms. This is done using the contingency table used to calculate NMI, which is just a table with the number of rows equalling the number of truth classes, and the number of columns equalling the number of clusters. Each cell in the table is a count of the number of observations assigned to the corresponding class and cluster labels. We turn this into a confusion matrix by merging each cluster-column to class-columns indicated by their row (class) which has the maximum count. Some classes will have zero counts, and multiple clusters may be merged into one class. We believe this is entirely unbiased, but may heavily penalise the clustering results in situations where no clusters map to a class. Also, trivial clustering solutions may be rewarded, i.e., when many clusters are found there is a greater chance they will be merged into the correct classes. It is worth noting that NMI does not suffer from these problems. We do not use training or test sets since no labels are used by the algorithm. Also there is no closed form solution for predicting labels on new data with the SCM and MCM.

For all datasets, the MCM has a truncation level $T_{trunc} = 30$ and the SCM $T_{trunc} = 100$. The SCM and MCM are also run ten times for each dataset with a random initialisation of $\mathbf{Y}$. The VDP, G-LDA, SCM and MCM code is all written in multi-threaded C++, though we only use one thread for most experiments to be strictly fair in our comparisons. All experiments were performed on a 2009 Core 2 Duo 3 GHz system with 6GB RAM.

6.1. MSRC

The first dataset considered is Microsoft's MSRC v2 dataset, which has both scene and object labels. We use the same 10 scene categories as [4, 6], with a total of 320 images (320×213 pixels). These images contained 15 segmented object categories, the "void" object category was not included. We found that 5×5 pixel un-normalised patches worked best for the ICA descriptors (with a dictionary of 50 bases).

The results for image clustering/classification are given in Table 1, with the line separating the unsupervised from the supervised algorithms. The VDP+ScSPM refers to running the



(a) Image clusters. (b) Segment clusters.

Figure 5: (a) A random selection of images from 8 of the 15 image clusters found by our proposed model on the MSRC dataset, (b) some of the (28) corresponding segment clusters. The image clusters have a normalised mutual information (NMI) score of 0.731, the segment clusters have an NMI of 0.580. No training or annotation data is used.

Table 1: Image performance for the MSRC dataset. More statistics for the MCM and SCM are shown in Figure 6. The VDP finds $T = 14$. #0 indicates the average number of unassigned classes or zeros on the diagonal of the confusion matrix. The horizontal line separates the unsupervised from (weakly) supervised algorithms.

Algorithm	**NMI**	**Acc.** (% (std.), #0)
MCM ($C_{w,s} = 0.4$, $C_{w,i} = 0.08$)	0.713 (0.023)	72.0 (3.3), 1.1
SCM ($C_{w,s} = 0.2$, $\theta = 1$)	0.652 (0.018)	63.5 (3.0), 2.1
VDP+ScSPM ($C_{w,i} = 0.02$)	0.636	56.69, 2
SC+ScSPM [46]	0.643 (0.002)	66.1 (1.6), 2.1
L^2-LEM-χ^2 [28] (dense SIFT BoW)	0.554 (0.018)	62.0 (2.7), 1.1
Du *et. al.* [4]	0.745	82.9
Du *et. al.* [4] LSBP	0.801	86.8
Li *et. al.* [6]	**0.820**	**89.06**

VDP with the image ScSPM based descriptors, and SC+ScSPM refers to self-tuning spectral clustering (SC) [46] using ScSPM features. For spectral clustering we use 10 random restarts, a 10 nearest neighbour sparse similarity matrix, and set the number of clusters to be the true number of classes. We also use self-tuning spectral clustering with $T = 10$ for the L^2-LEM-χ^2 algorithm with BoW features [28], but with a dense similarity matrix and a chi-square kernel.

The MCM performs substantially better for image clustering than all of the other unsupervised methods for this dataset, but does lag behind the weakly supervised methods of [4, 6]. However, the MCM still manages to achieve visually consistent image and segment clusters, see Figure 5.

Segment clustering performance was quantified on a per-segment basis, as opposed to per-pixel, which would have been too costly to evaluate for all images in these experiments. Also, the labelled segments provided were fairly coarse (this is especially true of the next dataset), and so we are more concerned with consistently recognising the same objects, as opposed to extracting them precisely. In order to assign a segment a ground-truth label, the mode of the pixels in the segment had to be of the label type. To quantify the algorithms' segment clustering performance, we ran them for an array of $C_{w,s}$ values. The results are summarised in Figure 6. We can see that the MCM consistently outperforms the VDP and G-LDA, wheras the SCM initially starts on par with the MCM, but then converges to the same performance as G-LDA. For this dataset scene-type context can improve performance over image-context and no context for segment clustering. We can also see the quality of the scene-types (image clusters) found has an effect on the quality of the object-types found.

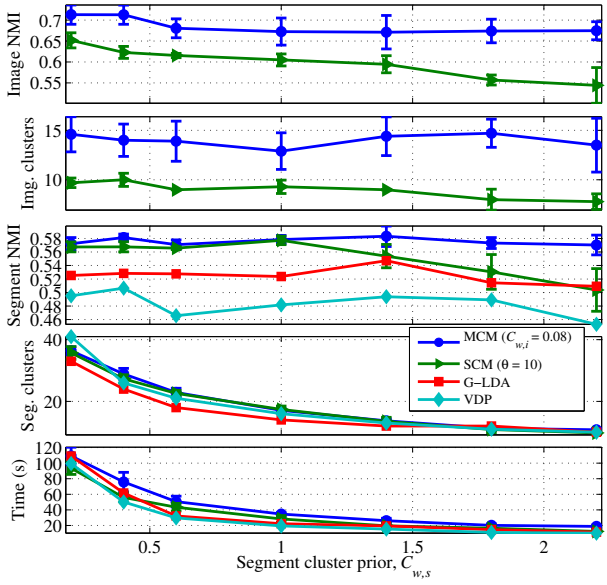


Figure 6: Segment performance on the MSRC v2 dataset. The MCM uses $C_{w,i} = 0.08$ for the image cluster prior. The VDP, G-LDA and SCM can only observe $\mathbf{x}_{jin}$ here.

We have only used a single mean-shift parameterisation that leads to over-segmentation for these results. It would seem prudent to cluster segments from multiple segmentation parameterisations simultaneously, as in [20, 26]. In this way, the probability of extracting the "true" object boundaries within a scene is increased. However, the MCM and SCM both assume that a scene-type is represented as a distribution of objects, so incorporating multiple segmentation results could bias scene discovery. We will leave multiple segmentation as the subject of future work, and for now rely on over-segmentation to achieve reasonable object boundary detection.

6.2. *LabelMe*

The next dataset we used was obtained from LabelMe [47]. It is comprised of 2688 images (256×256 pixels), with 8 scene classes. Here we found 7×7 un-normalised image patches worked best for the ICA descriptors (60 bases). The segment labels for this dataset were unconstrained in their categories, and so using the LabelMe Matlab toolbox, we combined all of the labels with 5 or more instances into 22 classes[1].The appearance

[1]The manifest file is located: www.daniel-steinberg.info/publications.html

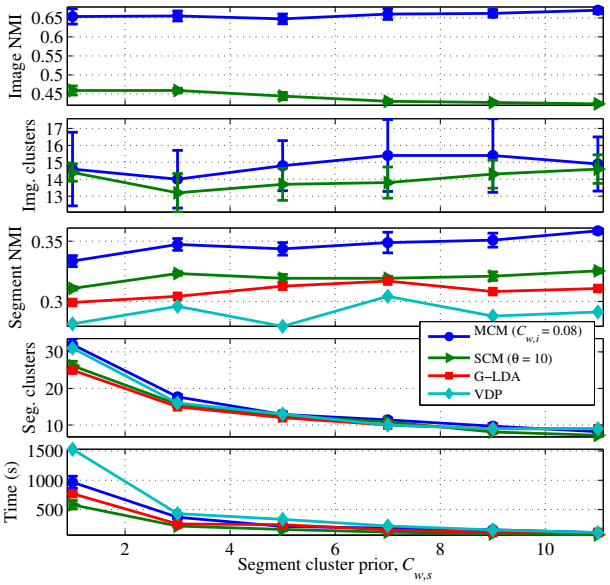


Figure 7: Segment performance for LabelMe. Again, the VDP, G-LDA and SCM can only observe $\mathbf{x}_{jin}$ here.

Table 2: Image performance for the LabelMe dataset for $C_{w,i} = 0.08$. The VDP finds $T = 8$.

Algorithm	**NMI**	**Acc.** (% (std.), #0)
MCM ($C_{w,s} = 11$, $C_{w,i} = 0.08$)	0.670 (0.009)	80.0 (2.8), 0.1
SCM ($C_{w,s} = 3$, $\theta = 10$)	0.459 (0.006)	58.4 (1.2), 0.5
VDP+ScSPM ($C_{w,i} = 0.08$)	0.708	82.3, 0
SC+ScSPM [46]	0.679 (0.017)	74.1 (3.5), 1.1
Li *et. al.* [6]	0.600	76.25
sLDA [2]	0.606	76
sLDA [2] (annots.)	0.606	76
DiscLDA+GC [14]	0.646	81
CD-BCC [7]	0.682	83.15
SVM + ScSPM [41]	0.6958	84.38
CA-TM [14]	**0.729**	**87**

of these object classes is far less constrained than the MSRC dataset.

Again we compare the unsupervised methods to state-of-the-art weakly, semi- and supervised methods in Table 2. Interestingly, the VDP performs best out of all of the unsupervised methods, it even outperforms a supervised support vector machine (SVM) using unmodified ScSPM features. The VDP is followed by spectral clustering and the MCM, which are within one standard deviation. The SCM performs very poorly on this dataset. All unsupervised methods apart from the SCM are quite competitive with the supervised methods on this dataset.

In this experiment it appears that the segment clusters are confounding the SCM image clustering. This would also explain the disparity between the VDP and the MCM, though the MCM is more robust than the SCM.

From Figure 7 we can see the MCM far outperforms the other unsupervised algorithms for segment clustering. The SCM marginally outperforms G-LDA, which both outperform the VDP. This demonstrates that object discovery can be improved by incorporating scene contextual information. Like in the MSRC dataset, we see that scene discovery performance can directly affect object discovery performance.

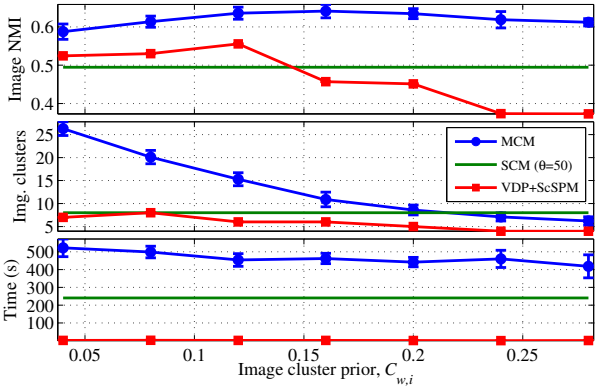


Figure 8: Image performance for the UIUC sports dataset. $C_{w,s} = 1$ was used for the MCM and SCM. Only the mean value for the SCM for the segment cluster prior is shown since it does not cluster image descriptors and so does not make use of the $C_{w,i}$ prior.

It is worth making some remarks on the visual effect on the clusters as we change the $C_{w,i}$ and $C_{w,s}$ parameters. Typically smaller settings of these parameters lead to more clusters (though after a point we may get fewer clusters once more). Usually larger clusters would be split fairly evenly into smaller clusters unless the images comprising a cluster were quite distinct. This is a consequence of the free energy objective, which tends to only make small clusters if the data-fitting terms overcome the complexity penalties.

6.3. UIUC Sports

The final standard dataset is the UIUC sports dataset used by [2–4, 6, 14]. This dataset depicts 8 types of sporting events and has 1579 images. To be consistent, we limit the maximum dimension of the images to be 320 pixels. Unfortunately no segment labels were available for this dataset. We use the same segment descriptor settings as the MSRC dataset. Results for scene recognition are presented in Table 3. Note that the algorithm from [4] is also fully unsupervised for this dataset.

Image classification in this dataset is more difficult than the others presented so far, as evident in the lower NMI and classification scores. Despite this, somewhat surprisingly, the MCM is one of the best performing algorithms on this dataset, including the supervised algorithms. We have also compared the MCM and VDP for image clustering with varying $C_{w,i}$ in Figure 8. There are no groups in this dataset, so G-LDA would be very similar to the VDP in performance. We have also plotted the performance of the SCM with $\theta = 50$ (average of 10 runs), which was found to give the best results. The MCM far outperforms the VDP, and is far more consistent in its performance across the range of $C_{w,i}$ chosen. However, it does take longer to find image clusters than both the SCM and the VDP. An example of the image and segment clusters found by the MCM is shown in Figure 9.

Table 3: Image performance for UIUC sport events. The VDP finds $T = 6$. The MCM finds mean $K = 30.2$, and the SCM $K = 20$, for other statistics see Figure 8.

Algorithm	NMI	Acc. (% (std.), #0)
MCM ($C_{w,s} = 1, C_{w,i} = 0.16$)	**0.641** (0.018)	74.1 (1.5), 1
SCM ($C_{w,s} = 1, \theta = 50$)	0.495 (0.008)	63.3 (1.5), 0.6
VDP+ScSPM ($C_{w,i} = 0.12$)	0.557	63.4, 2
SC+ScSPM [46]	0.429 (0.02)	58.9 (2.4), 1.1
Du *et. al.* [4] no LSBP	0.389	60.5
Du *et. al.* [4] LSBP	0.418	63.5
Li *et. al.* [3]	0.276	54
sLDA [2] (annots.)	0.438	66
sLDA [2]	0.446	65
Li *et. al.* [6]	0.466	69.11
DiscLDA+GC [14]	0.506	70
SVM+ScSPM [41]	0.549	72.9
CD-BCC [7]	0.556	75.15
CA-TM [14]	0.592	**78**

(a) Image clusters. (b) Segment clusters.

Figure 9: (a) A random selection of images from 5 of the 10 image clusters found by the MCM on the UIUC dataset, and (b) some of the (30) corresponding segment clusters. Here $C_{w,i} = 0.16$ and $C_{w,s} = 1$, the image clusters have a NMI score of 0.652 and an estimated accuracy of 74.0%.

6.4. Large AUV dataset

The last dataset we use is a novel dataset containing images of various underwater habitats obtained by an AUV from $J = 12$ deployments off of the east coast of Tasmania, Australia [48]. This dataset has 100,647 downward looking stereo pair images taken from an altitude of 2m. The monochrome image of the pair is used for the ScSPM descriptors, and the colour for the ICA segment descriptors. The images are reduced to 320×235 pixels (again to be consistent with the previous experiments). We used 5×5 pixels patches that had their DC components removed and were contrast normalised for both ICA dictionary learning (50 bases) and encoding. This helped with the illumination variations in this dataset.

This dataset has nine image classes: *fine sand*, *coarse sand*, *screw shell rubble* $\geq 50\%$, *screw shell rubble* $< 50\%$, *sand/reef interface*, *patch reef*, *low relief reef*, *high relief reef*, *Ecklonia (kelp)*. 6011 of these images are labelled. Many of these classes are quite visually similar so the labels have a small amount of noise. Exemplars of each class are shown in Figure 10. We do not use any segment labels for this dataset.

The entire 100,647 image dataset was clustered using the VDP, G-LDA, SCM and MCM. The 12 dives act as albums for G-LDA, the SCM and the MCM, and the SCM and MCM

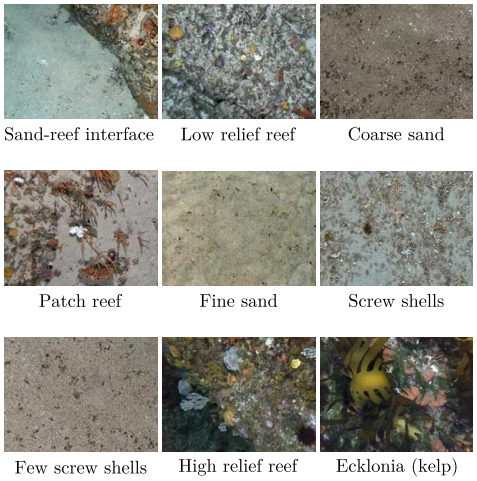


Figure 10: Exemplars of the nine AUV habitat classes.

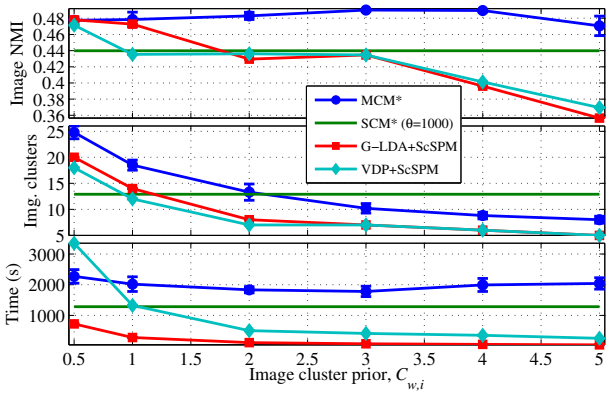
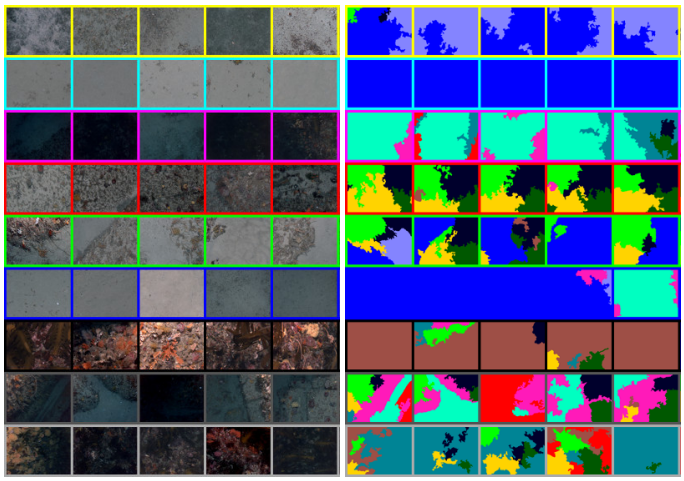


Figure 11: Image clustering performance for the Tasmania AUV dataset. $C_{w,s} = 350$ was used for the MCM and SCM. The (*) in the legend means the MCM and SCM used 8 cores of a Xeon (E5-4260) 2.2 GHz processor. The VDP and G-LDA used one core of a 3.0 GHz Core 2 Duo processor. Note the significant run-time difference between G-LDA and the VDP.

use $C_{w,s} = 350$. The VDP cannot take advantage of the groups in this dataset. Figure 11 summarises these results for varying $C_{w,i}$ (the 6011 labelled images were used for validation). The VDP, G-LDA and MCM all begin with similar image clustering performance, however the VDP and G-LDA rapidly decrease in performance with increasing $C_{w,i}$, whereas the MCM maintains, and even increases its performance. Again we just show the average of 10 runs with $\theta = 1000$ for the SCM since it does not use $C_{w,i}$. An example of some clusters found by the MCM are shown in Figure 12. Interestingly, the MCM has sufficient evidence to model the image vignetting and distortion artefacts induced by the air-acrylic-water interface of the camera housing. This is particularly evident in the fourth image cluster from the top (red).

Since the VDP and G-LDA only use image observations, they are run on the Core 2 Duo machine using one core. The SCM and MCM also have two million segments to cluster in addition to the images, and so they are run on eight Xeon E5-4260 2.2 GHz cores. It is interesting to note in Figure 11 that although the VDP and G-LDA only use one core, G-LDA is



(a) Random image cluster examples (b) Corresponding segment clusters

Figure 12: (a) A random selection of images from all of the 9 image clusters found by the MCM. Also shown in (b) are the corresponding segment clusters (11 in total). In this run, the image clusters achieved a NMI score of 0.499, and the segments a NMI score of 0.325. The priors used were $C_{w,i} = 0.27$ and $C_{w,s} = 18$.

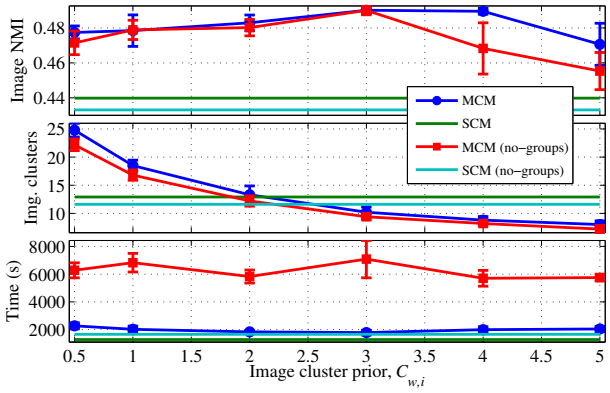


Figure 13: Comparison of modelling the AUV dives as separate groups compared to just one group on the AUV dataset. The MCM and SCM use $C_{w,s} = 350$ for the segment cluster prior, and both SCM variants use $\theta = 1000$.

consistently faster despite the two algorithms being very similar. We believe this is because modelling the different groups, or dives, in this dataset helps to expose separation between the clusters in the dataset. For instance, there are some dives that may exhibit only coarse sand while others may also exhibit screw shell rubble because of their location. So although these two classes may appear visually similar, the contextual information inherent in the dives' locations helps to distinguish these classes [39]. G-LDA can take advantage of this structure, which helps to simplify inference.

Like G-LDA, the MCM and SCM can also model groups. To verify if these algorithms can take advantage of this group structure we have compared them using $J = 12$ groups to only $J = 1$ group by concatenating the data from the dives. The results are shown in Figure 13. The MCM with $J = 1$ seems to perform worse than the $J = 12$ version for image clustering for some values of $C_{w,i}$. Also the SCM without groups performs marginally worse. The largest difference though is the MCM runtime with no groups. This runtime difference can be partially attributed to the way the MCM code is parallelised. The VBE step for the $\mathbf{Y}$ indicators (19) is parallelised over groups in the original MCM, and so the MCM without groups cannot take advantage of this. However, the VBE step for the segment indicators, $\mathbf{Z}$, has a larger computational cost (there are twenty times more segment observations than image observations), and is paralellised over images the same way in both MCM variants. So, it is unlikely the difference in the way the VBE-$\mathbf{Y}$ step is parallelised can entirely account for the three-fold run time increase. Part of this is likely attributable to the same cluster separation effect experience by G-LDA. The MCM with $J = 12$ incorporates G-LDA as part of its image model, where as when it has $J = 1$ it is more similar to the VDP at the image level.

6.5. Symmetric Dirichlet vs. Generalised Dirichlet

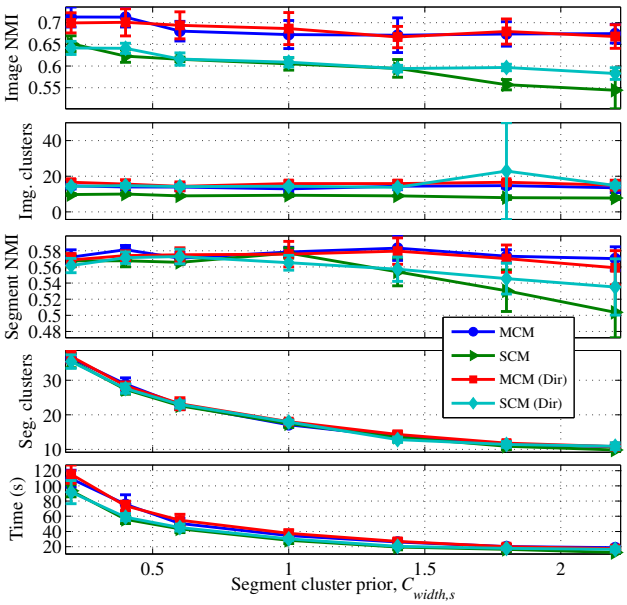
In section 3 we used a generalised Dirichlet prior over the group mixture weights, $\boldsymbol{\pi}_j$, for the SCM and MCM. Here we wish to empirically validate this choice of prior. For this purpose, we formulate variants of these two models with symmetric Dirichlet priors, $\boldsymbol{\pi}_j \sim \mathrm{Dir}(a)$, for comparison. These variants are run on all of the datasets mentioned previously, and the results are summarised in Figure 14.

As we can see from Figure 14, the symmetric Dirichlet prior versions of the SCM and MCM have very comparable NMI scores to their generalised Dirichlet counterparts. The SCM variants do show show slightly more variability in the UIUC and AUV datasets though. The MCM has almost the exact same results all-round, whereas the SCM with a symmetric Dirichlet prior finds more image clusters. In fact it appears to almost always find $T = T_{trunc} = 100$ image clusters for all but the small MSRC dataset. Even then it shows potential instability when looking at the variance at $C_{w,s} = 1.8$.

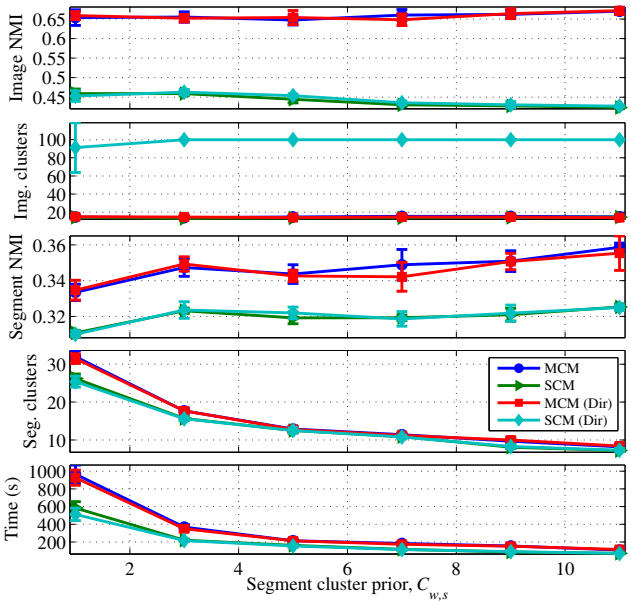
The generalised Dirichlet is a suitable choice of mixture weight prior for the SCM since it helps to control the number of image clusters found. This could be because it has twice the number of parameters as the Dirichlet prior, and so contributes more to the free energy model complexity penalty. This performance is also similar to the difference between LDA and HDPs when used for model selection in natural language processing [34]. The choice of prior does not seem to matter as much for the MCM, as the Gaussian image clusters contribute adequately to model complexity.

7. Conclusion

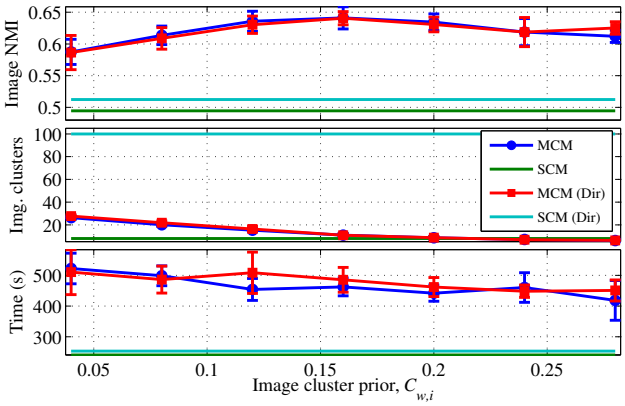
It has been long established that using discriminative visual features is essential for both unsupervised and supervised applications such as scene recognition, object detection, and scene understanding. In this work we have also shown that the choice of model structure has a large influence on results for scene understanding tasks in the absence of any semantic knowledge such as image tags, accompanying text, or image or object labels. We have also shown that with appropriate model structure and choice of visual features, unsupervised methods can



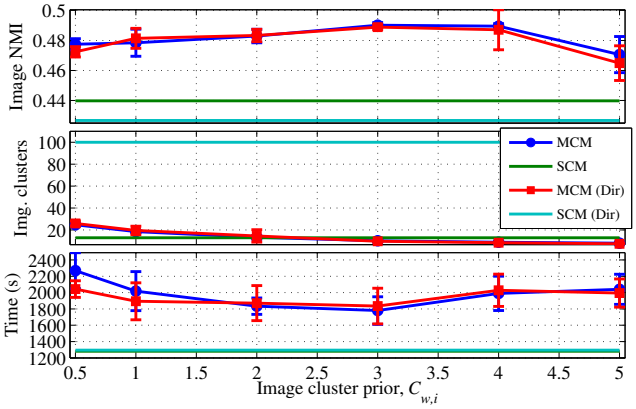
(a) MSRC v2, $C_{w,i} = 0.08$, $\theta = 1$



(b) LabelMe, $C_{w,i} = 0.08$, $\theta = 10$



(c) UIUC, $C_{w,s} = 1$, $\theta = 50$



(d) AUV, $C_{w,s} = 350$, $\theta = 1000$

Figure 14: Comparison of generalised Dirichlet vs. Dirichlet image cluster mixture priors for the SCM and MCM on the all of the previous datasets. The Dirichlet version of the SCM shows rather pathelogical behaviour for image clustering, in that it often finds $T = T_{trunc} = 100$ image clusters. The MCM variants are almost indistinguishable.

be competitive with weakly, semi, and supervised methods for scene understanding. From the experiments in section 6 we have also been able to conclude:

- Modelling albums or groups did not provide a large benefit to clustering performance. However algorithm run-time can be significantly reduced by additional parallelisation and using the structure of the groups to help expose the latent clusters.

- Scene-type context for segment clustering (SCM and MCM) helped performance more than image context (G-LDA), which is in turn better than no context (VDP). The better the discovered scene-types, the better the segment-clusters.

- The generalised Dirichlet prior on the group weights, $\boldsymbol{\pi}_j$, helps to control the number of Multinomial image clusters found in the SCM. However, it does not appear to affect models with Gaussian clusters, such as the MCM. This was also found to be the same for G-LDA.

- The Multinomial "bag-of-segments" representation for images used by the SCM does not perform as well as the mixture of Gaussian ScSPM representation in most cases. Whereas the ScSPM representation does not work well on small datasets, and can be sensitive to the choice of prior, $C_{w,i}$. The MCM combines the strengths of both of these complementary approaches, usually resulting in better and more robust image and segment clustering performance.

One direction for future work would be to model spatial-layout within the graphical models themselves, like in [14], while simultaneously modelling the co-occurrence between objects. Perhaps in this way joint object spatial and co-occurrence structure could be learned in an unsupervised manner.

8. Acknowledgements

This work is funded by the Australian Research Council, the New South Wales State Government, and the Integrated Marine Observing System. The authors acknowledge the providers of the datasets and those who released their code that was used in the validation of this work.

References

[1] D. M. Blei, M. I. Jordan, Modeling annotated data, in: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval, SIGIR '03, ACM, New York, NY, USA, 2003, pp. 127–134.

[2] C. Wang, D. Blei, L. Fei-Fei, Simultaneous image classification and annotation, in: Computer Vision and Pattern Recognition. CVPR. IEEE Conference on, IEEE, 2009, pp. 1903–1910.

[3] L.-J. Li, R. Socher, L. Fei-Fei, Towards total scene understanding: Classification, annotation and segmentation in an automatic framework, in: Computer Vision and Pattern Recognition (CVPR). IEEE Conference on, 2009, pp. 2036–2043.

[4] L. Du, L. Ren, D. Dunson, L. Carin, A Bayesian model for simultaneous image clustering, annotation and object segmentation, in: Advances in Neural Information Processing Systems, Vol. 22, 2009, pp. 486–494.

[5] R. Socher, L. Fei-Fei, Connecting modalities: Semi-supervised segmentation and annotation of images using unaligned text corpora, in: Computer Vision and Pattern Recognition. CVPR. IEEE Conference on, 2010, pp. 966–973.

[6] L. Li, M. Zhou, G. Sapiro, L. Carin, On the integration of topic modeling and dictionary learning, International Conference on Machine Learning.

[7] G. Irie, D. Liu, Z. Li, S.-F. Chang, A Bayesian approach to multimodal visual dictionary learning, in: Computer Vision and Pattern Recognition. CVPR. IEEE Conference on, IEEE, 2013.

[8] H. Attias, A variational Bayesian framework for graphical models, Advances in neural information processing systems 12 (1-2) (2000) 209–215.

[9] D. M. Blei, A. Y. Ng, M. I. Jordan, Latent Dirichlet allocation, The Journal of Machine Learning Research 3 (2003) 993–1022.

[10] D. Wulsin, S. Jensen, B. Litt, A Hierarchical Dirichlet process model with multiple levels of clustering for human EEG seizure modeling, in: Internation Conference on Machine Learning (ICML), 2012.

[11] A. Oliva, A. Torralba, Building the gist of a scene: The role of global image features in recognition, Progress in brain research 155 (2006) 23.

[12] A. Oliva, A. Torralba, The role of context in object recognition, Trends in Cognitive Sciences 11 (12) (2007) 520–527.

[13] L. Cao, L. Fei-Fei, Spatially coherent latent topic model for concurrent segmentation and classification of objects and scenes, in: Computer Vision. ICCV. IEEE 11th International Conference on, 2007, pp. 1–8.

[14] Z. Niu, G. Hua, X. Gao, Q. Tian, Context aware topic model for scene recognition, in: Computer Vision and Pattern Recognition. CVPR. IEEE Conference on, 2012, pp. 2743–2750.

[15] E. Sudderth, A. Torralba, W. Freeman, A. Willsky, Learning hierarchical models of scenes, objects, and parts, in: Computer Vision. ICCV. Tenth IEEE International Conference on, Vol. 2, 2005, pp. 1331–1338.

[16] L. Fei-Fei, L.-J. Li, What, where and who? telling the story of an image by activity classification, scene recognition and object categorization, in: R. Cipolla, S. Battiato, G. Farinella (Eds.), Computer Vision, Vol. 285 of Studies in Computational Intelligence, Springer Berlin Heidelberg, 2010, pp. 157–171.

[17] A. Torralba, K. Murphy, W. Freeman, M. Rubin, Context-based vision system for place and object recognition, in: Computer Vision. ICCV. Ninth IEEE International Conference on, Vol. 1, 2003, pp. 273–280.

[18] A. Torralba, K. P. Murphy, W. T. Freeman, Using the forest to see the trees: exploiting context for visual object detection and localization, Commun. ACM 53 (3) (2010) 107–114.

[19] M. J. Choi, J. Lim, A. Torralba, A. Willsky, Exploiting hierarchical context on a large database of object categories, in: Computer Vision and Pattern Recognition. CVPR. IEEE Conference on, 2010, pp. 129–136.

[20] Y. J. Lee, K. Grauman, Object-graphs for context-aware category discovery, in: Computer Vision and Pattern Recognition. CVPR. IEEE Conference on, IEEE, 2010, pp. 1–8.

[21] X. Wang, E. Grimson, Spatial latent Dirichlet allocation, Advances in Neural Information Processing Systems 20 (2007) 1577–1584.

[22] Q. An, C. Wang, I. Shterev, E. Wang, L. Carin, D. B. Dunson, Hierarchical kernel stick-breaking process for multi-task image analysis, in: Proceedings of the 25th international conference on Machine learning, ACM, 2008, pp. 17–24.

[23] E. B. Sudderth, M. I. Jordan, Shared segmentation of natural scenes using dependent Pitman-Yor processes, Advances in Neural Information Processing Systems 21 (2009) 1585–1592.

[24] B. Zhao, L. Fei-Fei, E. Xing, Image segmentation with topic random field, in: K. Daniilidis, P. Maragos, N. Paragios (Eds.), Computer Vision ECCV 2010, Vol. 6315 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2010, pp. 785–798.

[25] D. M. Steinberg, O. Pizarro, S. B. Williams, Synergistic clustering of image and segment descriptors for unsupervised scene understanding, in: Computer Vision (ICCV), 2013 IEEE International Conference on, 2013, pp. 3463–3470.

[26] B. C. Russell, W. T. Freeman, A. A. Efros, J. Sivic, A. Zisserman, Using multiple segmentations to discover objects and their extent in image collections, in: Computer Vision and Pattern Recognition. CVPR. IEEE Conference on, IEEE, 2006, pp. 1605–1614.

[27] C. Galleguillos, B. McFee, S. Belongie, G. Lanckriet, From region similarity to category discovery, in: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, IEEE, 2011, pp. 2665–2672.

[28] T. Tuytelaars, C. H. Lampert, M. B. Blaschko, W. Buntine, Unsupervised object discovery: A comparison, International Journal of Computer Vision 88 (2) (2010) 284–302.

[29] R. Gomes, M. Welling, P. Perona, Incremental learning of nonparametric Bayesian mixture models, in: Computer Vision and Pattern Recognition. CVPR. IEEE Conference on, IEEE, 2008, pp. 1–8.

[30] N. Bouguila, D. Ziou, A nonparametric Bayesian learning model: Application to text and image categorization, Advances in Knowledge Discovery and Data Mining (2009) 463–474.

[31] D. Dai, T. Wut, S.-C. Zhu, Discovering scene categories by information projection and cluster sampling, in: Computer Vision and Pattern Recognition (CVPR). IEEE Conference on, IEEE, 2010, pp. 3455–3462.

[32] K. Kurihara, M. Welling, N. Vlassis, Accelerated variational Dirichlet process mixtures, Advances in Neural Information Processing Systems 19 (2007) 761.

[33] T. S. Ferguson, A Bayesian analysis of some nonparametric problems, The Annals of Statistics 1 (2) (1973) 209–230.

[34] Y. W. Teh, M. I. Jordan, M. J. Beal, D. M. Blei, Hierarchical Dirichlet processes, Journal of the American Statistical Association 101 (476) (2006) 1566–1581.

[35] R. J. Connor, J. E. Mosimann, Concepts of independence for proportions with a generalization of the Dirichlet distribution, Journal of the American Statistical Association 64 (325) (1969) 194–206.

[36] T. T. Wong, Generalized Dirichlet distribution in Bayesian analysis, Applied Mathematics and Computation 97 (2-3) (1998) 165–181.

[37] H. Ishwaran, L. F. James, Gibbs sampling methods for stick-breaking priors, Journal of the American Statistical Association 96 (453) (2001) 161–173.

[38] C. M. Bishop, Pattern Recognition and Machine Learning, Springer Science+Business Media, Cambridge, UK, 2006.

[39] D. M. Steinberg, An unsupervised approach to modelling visual data, Ph.D. thesis, The University of Sydney (2013).

[40] M. J. Beal, Variational algorithms for approximate bayesian inference, Ph.D. thesis, University College London (2003).

[41] J. Yang, K. Yu, Y. Gong, T. Huang, Linear spatial pyramid matching using sparse coding for image classification, in: Computer Vision and Pattern Recognition. CVPR. IEEE Conference on, 2009, pp. 1794–1801.

[42] A. Coates, A. Ng, The importance of encoding versus training with sparse

coding and vector quantization, in: Proceedings of the 28th International Conference on Machine Learning, ICML '11, ACM, New York, NY, USA, 2011, pp. 921–928.
[43] A. Hyvärinen, E. Oja, Independent component analysis: algorithms and applications, Neural Networks 13 (45) (2000) 411–430.
[44] C. Christoudias, B. Georgescu, P. Meer, Synergism in low level vision, in: 16th International Conference on Pattern Recognition, Vol. 4, 2002, pp. 150–155 vol.4.
[45] A. Strehl, J. Ghosh, Cluster ensembles – a knowledge reuse framework for combining multiple partitions, Journal of Machine Learning Research 3 (2003) 583–617.
[46] L. Zelnik-Manor, P. Perona, Self-tuning spectral clustering, in: Advances in Neural Information Processing Systems, Vol. 17, 2004, pp. 1601–1608.
[47] B. C. Russell, A. Torralba, K. Murphy, W. Freeman, LabelMe: A database and web-based tool for image annotation, International Journal of Computer Vision 77 (2008) 157–173, 10.1007/s11263-007-0090-8.
[48] S. B. Williams, O. R. Pizarro, M. V. Jakuba, C. R. Johnson, N. S. Barrett, R. C. Babcock, G. A. Kendrick, P. D. Steinberg, A. J. Heyward, P. J. Doherty, I. Mahon, M. Johnson-Roberson, D. M. Steinberg, A. Friedman, Monitoring of benthic reference sites: using an autonomous underwater vehicle, Robotics Automation Magazine, IEEE 19 (1) (2012) 73–84.

Appendix A. Free Energy Objective Functions

In this appendix we give the full free energy objective functions for the SCM and MCM.

Appendix A.1. Simultaneous Clustering Model

$$\begin{aligned}\mathcal{F}_{\text{SCM}} = &\sum_{t=1}^{T}\mathbb{E}_{q_{\boldsymbol{\beta}}}\left[\log\frac{\text{Dir}(\boldsymbol{\beta}_t|\theta)}{q(\boldsymbol{\beta}_t)}\right]\\ &+\sum_{k=1}^{K}\mathbb{E}_{q_{\boldsymbol{\mu},\boldsymbol{\Lambda}}}\left[\log\frac{\mathcal{N}\left(\boldsymbol{\mu}_k|\mathbf{m},(\gamma\boldsymbol{\Lambda}_k)^{-1}\right)\mathcal{W}(\boldsymbol{\Lambda}_k|\boldsymbol{\Omega},\rho)}{q(\boldsymbol{\mu}_k,\boldsymbol{\Lambda}_k)}\right]\\ &+\sum_{j=1}^{J}\mathbb{E}_{q_{\boldsymbol{\pi}}}\left[\log\frac{\text{GDir}(\boldsymbol{\pi}_j|a,b)}{q(\boldsymbol{\pi}_j)}\right]\\ &+\sum_{j=1}^{J}\sum_{i=1}^{Ij}\mathbb{E}_{q}\left[\log\frac{\text{Categ}(y_{ji}|\boldsymbol{\pi}_j)}{q(y_{ji})}\right]\\ &+\sum_{j=1}^{J}\sum_{i=1}^{Ij}\sum_{n=1}^{N_{ji}}\sum_{t=1}^{T}\mathbb{E}_{q}\left[\log\frac{\text{Categ}(z_{jin}|\boldsymbol{\beta}_t)^{\mathbf{1}[y_{ji}=t]}}{q(z_{jin})}\right]\\ &+\sum_{j=1}^{J}\sum_{i=1}^{Ij}\sum_{n=1}^{N_{ji}}\sum_{k=1}^{K}\mathbb{E}_{q}\left[\log\frac{\mathcal{N}(\mathbf{x}_{jin}|\boldsymbol{\mu}_k,\boldsymbol{\Lambda}_k)^{\mathbf{1}[z_{jin}=k]}}{q(z_{jin})}\right],\end{aligned}\tag{A.1}$$

The last three terms' expectations are with respect to all of the latent variables and parameters. These last three terms act like a data-fitting objectives, and the first three terms act as model complexity penalties. It is also worth noting that $\mathbb{E}_{q_y}[\mathbf{1}[y_{ji}=t]] = q(y_{ji=t})$, and similarly for z_{jin}.

Appendix A.2. Multiple-source Clustering Model

$$\begin{aligned}\mathcal{F}_{\text{MCM}} = &\sum_{t=1}^{T}\mathbb{E}_{q_{\boldsymbol{\beta}}}\left[\log\frac{\text{Dir}(\boldsymbol{\beta}_t|\theta)}{q(\boldsymbol{\beta}_t)}\right]\\ &+\sum_{t=1}^{T}\mathbb{E}_{q_{\boldsymbol{\eta},\boldsymbol{\Psi}}}\left[\log\frac{\mathcal{N}\left(\boldsymbol{\eta}_t|\mathbf{h},(\delta\boldsymbol{\Psi}_t)^{-1}\right)\mathcal{W}(\boldsymbol{\Psi}_t|\boldsymbol{\Phi},\xi)}{q(\boldsymbol{\eta}_t,\boldsymbol{\Psi}_t)}\right]\\ &+\sum_{k=1}^{K}\mathbb{E}_{q_{\boldsymbol{\mu},\boldsymbol{\Lambda}}}\left[\log\frac{\mathcal{N}\left(\boldsymbol{\mu}_k|\mathbf{m},(\gamma\boldsymbol{\Lambda}_k)^{-1}\right)\mathcal{W}(\boldsymbol{\Lambda}_k|\boldsymbol{\Omega},\rho)}{q(\boldsymbol{\mu}_k,\boldsymbol{\Lambda}_k)}\right]\\ &+\sum_{j=1}^{J}\mathbb{E}_{q_{\boldsymbol{\pi}}}\left[\log\frac{\text{GDir}(\boldsymbol{\pi}_j|a,b)}{q(\boldsymbol{\pi}_j)}\right]\\ &+\sum_{j=1}^{J}\sum_{i=1}^{Ij}\mathbb{E}_{q}\left[\log\frac{\text{Categ}(y_{ji}|\boldsymbol{\pi}_j)}{q(y_{ji})}\right]\\ &+\sum_{j=1}^{J}\sum_{i=1}^{Ij}\sum_{t=1}^{T}\mathbb{E}_{q}\left[\log\frac{\mathcal{N}(\mathbf{w}_{ji}|\boldsymbol{\eta}_t,\boldsymbol{\Psi}_t)^{\mathbf{1}[y_{ji}=t]}}{q(y_{ji})}\right]\\ &+\sum_{j=1}^{J}\sum_{i=1}^{Ij}\sum_{n=1}^{N_{ji}}\sum_{t=1}^{T}\mathbb{E}_{q}\left[\log\frac{\text{Categ}(z_{jin}|\boldsymbol{\beta}_t)^{\mathbf{1}[y_{ji}=t]}}{q(z_{jin})}\right]\\ &+\sum_{j=1}^{J}\sum_{i=1}^{Ij}\sum_{n=1}^{N_{ji}}\sum_{k=1}^{K}\mathbb{E}_{q}\left[\log\frac{\mathcal{N}(\mathbf{x}_{jin}|\boldsymbol{\mu}_k,\boldsymbol{\Lambda}_k)^{\mathbf{1}[z_{jin}=k]}}{q(z_{jin})}\right],\end{aligned}\tag{A.2}$$

The last four terms' expectations are with respect to all of the latent variables and parameters. Now the last four terms act like a data-fitting objectives and the first four terms act as model complexity penalties.

Appendix B. Variational Expectations

Appendix B.1. Dirichlet Distribution

The Categorical distribution is most often used as the likelihood of the Dirichlet distribution in this paper,

$$\text{Categ}(z_{jin}|\boldsymbol{\beta}_t) = \prod_{k=1}^{K}\beta_{tk}^{\mathbf{1}[z_{jin}=k]}.\tag{B.1}$$

Here $\mathbf{1}[\cdot]$ is an indicator function, and evaluates to 1 when the condition in the brackets is true, and 0 otherwise. The corresponding Dirichlet prior has the form,

$$\text{Dir}(\boldsymbol{\beta}_t|\theta) = \frac{\Gamma(K\cdot\theta)}{\Gamma(\theta)^K}\prod_{k=1}^{K}\beta_{tk}^{\theta-1},\tag{B.2}$$

here $\Gamma(\cdot)$ is a Gamma function. This is a symmetric Dirichlet prior, another a commonly used parameterisation is $\text{Dir}(\boldsymbol{\beta}_t|\theta/K)$.

Appendix B.1.1. Expectations over the likelihood

The log Categorical expectation under a generalised Dirichlet is,

$$\begin{aligned}\mathbb{E}_{q_{\boldsymbol{\beta}}}[\log p(z_{jin}=k|\boldsymbol{\beta}_t)] &= \mathbb{E}_{q_\beta}[\log\beta_{tk}]\\ &= \Psi\left(\tilde{\theta}_{tk}\right) - \Psi\left(\sum_k\tilde{\theta}_{tk}\right),\end{aligned}\tag{B.3}$$

where $\Psi(\cdot)$ is a Digamma function, and $\tilde{\theta}_{tk}$ is from (20).

Appendix B.1.2. Free energy expectations

The expectations of the model complexity penalty terms are,

$$\begin{aligned}&\mathbb{E}_{q_{\boldsymbol{\beta}}}\left[\log\frac{\text{Dir}(\boldsymbol{\beta}_t|\theta)}{q(\boldsymbol{\beta}_t)}\right] = \log\Gamma(K\cdot\theta) - \log\Gamma\left(\sum_{k=1}^{K}\tilde{\theta}_{tk}\right)\\ &+\sum_{k=1}^{K}\log\Gamma\left(\tilde{\theta}_{tk}\right) - K\log\Gamma(\theta) - \sum_{k=1}^{K}\left(\tilde{\theta}_{tk}-\theta\right)\mathbb{E}_{q_\beta}[\log\beta_{tk}],\end{aligned}\tag{B.4}$$

where $\mathbb{E}_{q_\beta}[\log \beta_{tk}]$ is from Equation B.3.

Appendix B.2. Generalised Dirichlet Distribution

The Categorical distribution is most often used as the likelihood of the Generalised Dirichlet distribution in this paper, see Equation B.1. The generalised Dirichlet prior on the mixture weights, $\mathrm{GDir}(\boldsymbol{\pi}_j|a,b)$ is parameterised in (15), and made use of the Beta distribution over stick-lengths,

$$\mathrm{Beta}(v_{jt}|a,b) = \frac{\Gamma(a+b)}{\Gamma(a)\,\Gamma(b)} v_{jt}^{a-1}\left(1-v_{jt}\right)^{b-1}. \tag{B.5}$$

Appendix B.2.1. Expectations over the likelihood

The log Categorical expectation under a generalised Dirichlet is,

$$\begin{aligned}\mathbb{E}_{q_{\boldsymbol{\pi}}}[\log p(z_{jin}=k|\boldsymbol{\pi}_j)] &= \mathbb{E}_{q_\pi}[\log \pi_{jt}] \\ &= \mathbb{E}_{q_v}[\log v_{jt}] + \sum_{s=1}^{t-1}\mathbb{E}_{q_v}[\log(1-v_{js})], \end{aligned}\tag{B.6}$$

where,

$$\mathbb{E}_{q_v}[\log v_{jt}] = \begin{cases}\Psi(\tilde{a}_{jt}) - \Psi\left(\tilde{a}_{jt}+\tilde{b}_{jt}\right) & \text{if } t<T \\ 0 & \text{if } t=T,\end{cases}\tag{B.7}$$

$\tilde{a}_{jt}$ and $\tilde{b}_{jt}$ are given in (20), and

$$\mathbb{E}_{q_v}[\log(1-v_{jt})] = \Psi\left(\tilde{b}_{jt}\right) - \Psi\left(\tilde{a}_{jt}+\tilde{b}_{jt}\right) \quad \text{if } t<T. \tag{B.8}$$

Appendix B.2.2. Free energy expectations

The expectations of the model complexity penalty terms can be factorised,

$$\mathbb{E}_{q_{\boldsymbol{\pi}}}\left[\log \frac{\mathrm{GDir}(\boldsymbol{\pi}_j|a,b)}{q(\boldsymbol{\pi}_j)}\right] = \sum_{t=1}^{T-1}\mathbb{E}_{q_\pi}\left[\log \frac{p(\pi_{jt}|a,b)}{q(\pi_{jt})}\right], \tag{B.9}$$

where

$$\begin{aligned}\mathbb{E}_{q_\pi}\left[\log \frac{p(\pi_{jt}|a,b)}{q(\pi_{jt})}\right] &= -\left(\tilde{a}_{jt}-a\right)\mathbb{E}_{q_v}[\log v_{jt}] \\ &\quad -\left(\tilde{b}_{jt}-b\right)\mathbb{E}_{q_v}[\log(1-v_{jt})] + \log\Gamma(\tilde{a}_{jt}) \\ &\quad -\log\Gamma(a) + \log\Gamma\left(\tilde{b}_{jt}\right) - \log\Gamma(b) \\ &\quad -\log\Gamma\left(\tilde{a}_{jt}+\tilde{b}_{jt}\right) + \log\Gamma(a+b). \end{aligned}\tag{B.10}$$

The free energy penalty term over the weights in Equation B.10 only sums to $T-1$ (degrees of freedom).

Appendix B.3. Gaussian-Wishart Distribution

Gaussian distributions are often used to describe segment clusters[2] in this paper, which take the form,

$$\begin{aligned}\mathcal{N}\left(\mathbf{x}_{jin}|\boldsymbol{\mu}_k,\boldsymbol{\Lambda}_k^{-1}\right) &= \frac{|\boldsymbol{\Lambda}_k|^{1/2}}{(2\pi)^{D/2}} \\ &\quad\times \exp\left\{-\frac{1}{2}\left(\mathbf{x}_{jin}-\boldsymbol{\mu}_k\right)^\top \boldsymbol{\Lambda}_k\left(\mathbf{x}_{jin}-\boldsymbol{\mu}_k\right)\right\}. \end{aligned}\tag{B.11}$$

A Gaussian-Wishart prior is placed over the parameters,

$$\begin{aligned}\mathcal{N}\left(\boldsymbol{\mu}_k|\mathbf{m},(\gamma\boldsymbol{\Lambda}_k)^{-1}\right) &= \frac{|\gamma\boldsymbol{\Lambda}_k|^{1/2}}{(2\pi)^{D/2}} \\ &\quad\times \exp\left\{-\frac{\gamma}{2}\left(\boldsymbol{\mu}_k-\mathbf{m}\right)^\top \boldsymbol{\Lambda}_k\left(\boldsymbol{\mu}_k-\mathbf{m}\right)\right\}, \end{aligned}\tag{B.12}$$

$$\begin{aligned}\mathcal{W}(\boldsymbol{\Lambda}_k|\boldsymbol{\Omega},\rho) &= \frac{|\boldsymbol{\Lambda}_k|^{(\rho-D-1)/2}}{2^{\rho D/2}\,|\boldsymbol{\Omega}|^{\rho/2}\,\Gamma_D\left(\frac{\rho}{2}\right)} \\ &\quad\times \exp\left\{-\frac{1}{2}\mathrm{Tr}\left(\boldsymbol{\Omega}^{-1}\boldsymbol{\Lambda}_k\right)\right\}, \end{aligned}\tag{B.13}$$

where $\Gamma_D(\cdot)$ is a multivariate Gamma function,

$$\Gamma_D\left(\frac{\rho}{2}\right) = \pi^{D(D-1)/4}\prod_{d=1}^{D}\Gamma\left(\frac{\rho+1-d}{2}\right). \tag{B.14}$$

Appendix B.3.1. Expectations over the likelihood

The log Gaussian expectation under a Gaussian-Wishart prior is,

$$\begin{aligned}\mathbb{E}_{q_{\boldsymbol{\mu},\boldsymbol{\Lambda}}}\left[\log\mathcal{N}\left(\mathbf{x}_{jin}|\boldsymbol{\mu}_k,\boldsymbol{\Lambda}_k^{-1}\right)\right] &= \frac{1}{2}\mathbb{E}_{q_{\boldsymbol{\Lambda}}}[\log|\boldsymbol{\Lambda}_k|] \\ &\quad -\frac{D}{2\tilde{\gamma}_k} - \frac{\tilde{\rho}_k}{2}\left(\mathbf{x}_{jin}-\tilde{\mathbf{m}}_k\right)^\top\tilde{\boldsymbol{\Omega}}_k\left(\mathbf{x}_{jin}-\tilde{\mathbf{m}}_k\right), \end{aligned}\tag{B.15}$$

where

$$\mathbb{E}_{q_{\boldsymbol{\Lambda}}}[\log|\boldsymbol{\Lambda}_k|] = \sum_{d=1}^{D}\Psi\left(\frac{\tilde{\rho}_k+1-d}{2}\right) + D\log 2 + \log\left|\tilde{\boldsymbol{\Omega}}_k\right|. \tag{B.16}$$

All of the posterior parameters $(\tilde{\cdot})$ are from (21).

Appendix B.3.2. Free energy expectations

The expectations of the model complexity penalty terms are,

$$\begin{aligned}&\mathbb{E}_{q_{\boldsymbol{\mu},\boldsymbol{\Lambda}}}\left[\log\frac{\mathcal{N}\left(\boldsymbol{\mu}_k|\mathbf{m},(\gamma\boldsymbol{\Lambda}_k)^{-1}\right)\mathcal{W}(\boldsymbol{\Lambda}_k|\boldsymbol{\Omega},\rho)}{q(\boldsymbol{\mu}_k,\boldsymbol{\Lambda}_k)}\right] = \\ &-\frac{D}{2}\left(\frac{\gamma}{\tilde{\gamma}_k} - \log\frac{\gamma}{\tilde{\gamma}_k} - \tilde{\rho}_k - 1\right) - \frac{\rho}{2}\left(\log|\boldsymbol{\Omega}| - \log|\tilde{\boldsymbol{\Omega}}_k|\right) \\ &-\frac{\tilde{\rho}_k}{2}\mathrm{Tr}\left(\boldsymbol{\Omega}^{-1}\tilde{\boldsymbol{\Omega}}_k\right) - \frac{\tilde{\rho}_k\gamma}{2}\left(\tilde{\mathbf{m}}_k-\mathbf{m}\right)^\top\tilde{\boldsymbol{\Omega}}_k\left(\tilde{\mathbf{m}}_k-\mathbf{m}\right)\end{aligned}$$

[2] These equations are almost the same for the image clusters, so they are omitted.

$$- \sum_{d=1}^{D} \left(\frac{N_k}{2} \Psi\left(\frac{\tilde{\rho}_k + 1 - d}{2} \right) + \log \Gamma\left(\frac{\rho + 1 - d}{2} \right) - \log \Gamma\left(\frac{\tilde{\rho}_k + 1 - d}{2} \right) \right). \quad \text{(B.17)}$$

Appendix C. Model Selection Heuristic

The greedy splitting heuristic is based on two criteria. The first is the approximate free energy contribution of the segment cluster parameters and segment observations to be split. The second is how many split attempts have been tried for the segment cluster and not been accepted previously. The cluster split attempts are ordered by (a) least number of previous split attempts for the clusters, then (b) clusters with more free energy contribution. The first attempt that reduces model free energy is accepted. The approximate contribution to free energy is formulated from the heuristic,

$$\hat{\mathcal{F}}_k = \mathbb{E}_{q_{\boldsymbol{\mu},\boldsymbol{\Lambda}}} \left[\log \frac{\mathcal{N}\left(\boldsymbol{\mu}_k | \mathbf{m}, (\gamma \boldsymbol{\Lambda}_k)^{-1}\right) \mathcal{W}(\boldsymbol{\Lambda}_k | \boldsymbol{\Omega}, \rho)}{q(\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k)} \right] + \sum_{j=1}^{J} \sum_{i=1}^{I_j} \sum_{n=1}^{N_{ji}} q(z_{jin} = k)\, \mathcal{L}_{z_{jin}=k} \quad \text{(C.1)}$$

where $\mathcal{L}_{z_{jin}=k}$ is the mixture likelihood of observation $\mathbf{x}_{jin}$ under segment cluster k (including the effect of the mixture weights). This likelihood is weighted by the observation's probabilistic membership to cluster k. For the SCM and MCM the exact form of this heuristic is,

$$\hat{\mathcal{F}}_k = \mathbb{E}_{q_{\boldsymbol{\mu},\boldsymbol{\Lambda}}} \left[\log \frac{\mathcal{N}\left(\boldsymbol{\mu}_k | \mathbf{m}, (\gamma \boldsymbol{\Lambda}_k)^{-1}\right) \mathcal{W}(\boldsymbol{\Lambda}_k | \boldsymbol{\Omega}, \rho)}{q(\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k)} \right] + \sum_{j=1}^{J} \sum_{i=1}^{I_j} \sum_{n=1}^{N_{ji}} q(z_{jin} = k)\, \mathbb{E}_{q_{\boldsymbol{\mu},\boldsymbol{\Lambda}}} \left[\log \mathcal{N}\left(\mathbf{x}_{jin} | \boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k^{-1}\right) \right]. \quad \text{(C.2)}$$

A cluster weight term was not included in Equation C.2 because a corresponding term of opposite sign existed in the last term in Equation A.2, and adding it would nullify its effect in the overall model free energy. Whereas the heuristic for G-LDA applied to segment clustering is,

$$\hat{\mathcal{F}}_k = \mathbb{E}_{q_{\boldsymbol{\mu},\boldsymbol{\Lambda}}} \left[\log \frac{\mathcal{N}\left(\boldsymbol{\mu}_k | \mathbf{m}, (\gamma \boldsymbol{\Lambda}_k)^{-1}\right) \mathcal{W}(\boldsymbol{\Lambda}_k | \boldsymbol{\Omega}, \rho)}{q(\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k)} \right] + \sum_{i=1}^{I} \sum_{n=1}^{N_i} q(z_{in} = k) \left[\mathbb{E}_{q_\pi} [\log \pi_{ik}] + \mathbb{E}_{q_{\boldsymbol{\mu},\boldsymbol{\Lambda}}} \left[\log \mathcal{N}\left(\mathbf{x}_{in} | \boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k^{-1}\right) \right] \right], \quad \text{(C.3)}$$

which includes a cluster weight term.

The observations belonging to a Gaussian (with $q(z_{jin} = k) > 0.5$) are split in a direction perpendicular to its principal Eigenvector. This split is refined by iterating the VBE and VBM steps on only these observations. The algorithm is summarised in Algorithm 1 for the MCM, the SCM is similar but does not include $\mathbf{W}$. The expected model free energy, $\mathbb{E}[\mathcal{F}_{split,k}]$ is acquired by running variational Bayes for one iteration, with the new split, using all of the segment observations. To our knowledge this is the first time a split tally has been used in a cluster splitting heuristic. It was found to significantly reduce the run time of the algorithm and improve results over just using approximate free energy to guide the greedy search. This greedy cluster splitting heuristic often less than halved the run time of the total algorithm compared to the exhaustive cluster splitting heuristic. This speed-up was even more pronounced for the larger datasets. It also managed to maintain good clustering results compared to the exhaustive heuristic.

Algorithm 1: The MCM greedy model selection heuristic

Data: Observations $\mathbf{W}, \mathbf{X}$
Result: Assignments $q(\mathbf{Y})$ and $q(\mathbf{Z})$ and posterior hyper-parameters $\tilde{\boldsymbol{\Xi}}$

$\boldsymbol{\Xi} \leftarrow$ `CreatePriors()`;
$q(\mathbf{Y}) \leftarrow$ `RandomLabels`$(T_{trunc} = 30)$;
$q(\mathbf{Z}) \leftarrow \{\{\mathbf{1}\}_{i=1}^{I_j}\}_{j=1}^{J}$; `// initialises with` $K = 1$
splittally $\leftarrow \{0\}_{k=1}^{K}$;

repeat
  $q(\mathbf{Y}), q(\mathbf{Z}), \tilde{\boldsymbol{\Xi}}, \mathcal{F} \leftarrow$ `VB`$(\mathbf{X}, q(\mathbf{Y}), q(\mathbf{Z}), \boldsymbol{\Xi})$;
  splitorder $\leftarrow$ `GreedySort`$(\mathbf{W}, \mathbf{X}, q(\mathbf{Z}), G,$ splittally$)$; `// sequence`
  foreach $k \in$ splitorder **do**
    $\mathbf{X}_{split,k} \leftarrow \{\mathbf{x}_{jin} \in \mathbf{X} : q(z_{jin} = k) > 0.5\}$;
    $q(\mathbf{Z}_{split,k}) \leftarrow$ `ClusterSplit`$(\mathbf{X}_{split,k})$;
    $q(\mathbf{Z}_{split,k}) \leftarrow$ `VB`$(\mathbf{W}, \mathbf{X}_{split,k}, \{\mathbf{1}\}_{j=1}^{J}, q(\mathbf{Z}_{split,k}), \boldsymbol{\Xi})$;
    $q(\mathbf{Z}_{aug,k}) \leftarrow$ `AugmentLabels`$(q(\mathbf{Z}), q(\mathbf{Z}_{split,k}))$;
    $\mathbb{E}[\mathcal{F}_{split,k}] \leftarrow$ `VB`$(\mathbf{W}, \mathbf{X}, q(\mathbf{Y}), q(\mathbf{Z}_{aug,k}), \boldsymbol{\Xi})$; `// 1 iter.`
    if $\mathcal{F} > \mathbb{E}[\mathcal{F}_{split,k}]$ **then**
      $q(\mathbf{Z}) \leftarrow q(\mathbf{Z}_{aug,k})$;
      splittally$_k \leftarrow 0$;
      splittally$_{K+1} \leftarrow 0$;
      foundsplit $\leftarrow$ *true*;
      break;
    else
      splittally$_k \leftarrow$ splittally$_k + 1$;
      foundsplit $\leftarrow$ *false*;
until foundsplit $=$ *false*;

$q(\mathbf{Y}) \leftarrow$ `PruneEmptyClusters`$(q(\mathbf{Y}))$;