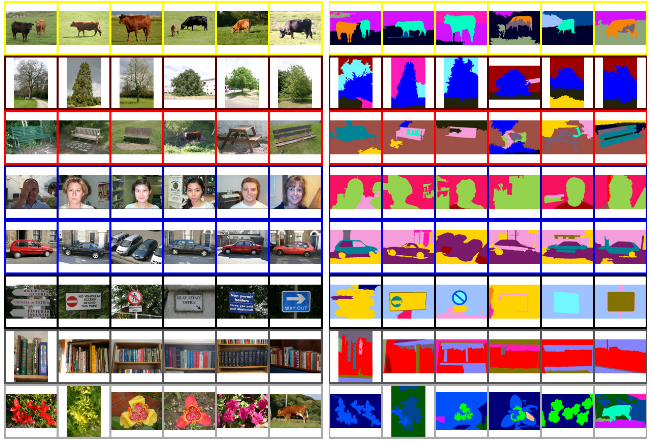
Synergistic Clustering of Image and Segment Descriptors for Unsupervised Scene Understanding

Daniel M. Steinberg, Oscar Pizarro, Stefan B. Williams
Australian Centre for Field Robotics, The University of Sydney
Sydney, NSW, 2006
{d.steinberg, o.pizarro, stefanw}@acfr.usyd.edu.au

Abstract

With the advent of cheap, high fidelity, digital imaging systems, the quantity and rate of generation of visual data can dramatically outpace a humans ability to label or annotate it. In these situations there is scope for the use of unsupervised approaches that can model these datasets and automatically summarise their content. To this end, we present a totally unsupervised, and annotation-less, model for scene understanding. This model can simultaneously cluster whole-image and segment descriptors, thereby forming an unsupervised model of scenes and objects. We show that this model outperforms other unsupervised models that can only cluster one source of information (image or segment) at once. We are able to compare unsupervised and supervised techniques using standard measures derived from confusion matrices and contingency tables. This shows that our unsupervised model is competitive with current supervised and weakly-supervised models for scene understanding on standard datasets. We also demonstrate our model operating on a dataset with more than 100,000 images collected by an autonomous underwater vehicle.



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

Figure 1. (a) A random selection of images from 8 of the 15 image clusters found by our proposed model on the MSRC dataset and (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.

1. Introduction

In many applications, the quantity and rate at which visual data is collected can far outpace a humans ability to label or annotate even a small percentage of it. One example of this is the collection of scientific visual data by autonomous agents such as planetary rovers, unmanned air vehicles (UAVs), or autonomous underwater vehicles (AUVs). Unsupervised "scene understanding" algorithms could summarise this data in the absence of any annotations. A human expert would then only need to view these summaries before directing their attention to relevant subsets of the data for subsequent analysis.

Scene understanding is an active area of research in computer vision. It refers to frameworks that incorporate and model multiple sources of visual, annotation or other information to improve some visual inference task. An early attempt proposed in [2] extends latent Dirichlet allocation (LDA) so both visual and textual data can be used for inferring image tags in untagged images. Subsequent research, [4, 7, 13, 15, 21, 25] (amongst others) present Bayesian models that combine multiple bag-of-words based image representations to 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 [7, 13, 25]. We refer to these models as "weakly-supervised". Some of these models also explicitly model the spatial layout of scenes [15, 21], or may make use of a non-parametric process to enforce segment-label contiguity [7]. An interesting model is presented in [12], which eschews a bag-of-words representation in favour of learning a sparse-code based image representation, while also simultaneously modelling scenes and annotations.

Some of the aforementioned models can also be used in a fully unsupervised setting, where no annotation or label data is available. However, these models may operate in a reduced capacity. For instance [4] loses its ability to classify images while inferring objects. Generally unsupervised methods for understanding visual data have been given less focus than supervised and weakly-supervised methods. This is because many applications have at least some annotation data available (like *Flickr* photos). There has been research into unsupervised methods for object discovery [17, 19, 24, 11] and scene discovery [8, 14], which are related but distinct problems to scene understanding.

In this paper, we present a Bayesian graphical model specialised for truly unsupervised scene understanding applications. We refer to it as the *multiple-source clustering model* (MCM). It is able to model multiple albums of images at both scene and object levels without human supervision. Rather than relying on human-generated scene labels, it infers scene-types by clustering images. It uses a whole-image descriptor *as well as* a latent distribution of "object" types to represent images. These object-types are formed by simultaneously clustering image-segment descriptors – hence multiple-source. See Figure 1 for an example of the MCM's output.

The structure of our model is such that scene-types can influence the objects found in an image (we would likely find trees in a forest). This is conceptually similar to the work in [23], which is inspired by research on the human visual cortex [16]. Global visual features are used to understand the context of a scene without explicitly registering the individual objects that compose the image. This scene recognition provides context that aids the recognition of objects, which otherwise may be difficult to recognise in isolation. Also, in the MCM the co-occurrence and distribution of objects within an image can influence the type of scene it belongs to (cows and grass likely make a rural scene).

The primary contribution of this work is in showing that the MCM can synergistically cluster both image and segment descriptors. It outperforms unsupervised models that only consider one source of information. It is also competitive with weakly-supervised and supervised models for scene understanding. We quantify this by using standard measures that can be derived from the confusion matrices reported in the literature. Finally we show the MCM is readily scalable to larger datasets of the kind we would expect from a robot.

2. The Multiple-source Clustering Model

In this section we present the multiple-source clustering model (MCM). A graphical model of the MCM is presented in Figure 2, and it assumes there are J albums, or groups of images, indexed by j. Each of these albums has I_j images, indexed by i, which in turn contain N_{ji} non-overlapping segments or superpixels, indexed by n.

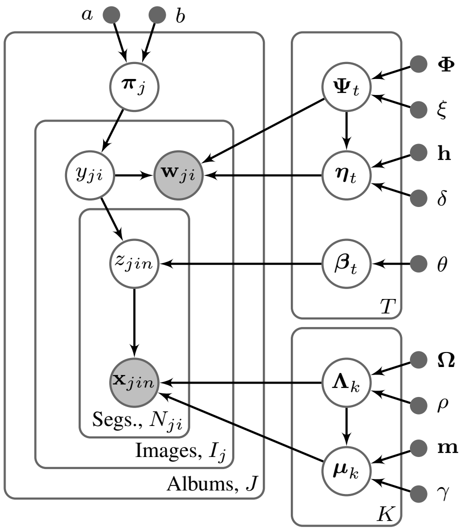


Figure 2. The Multiple-source clustering model. The shaded nodes are observable descriptors, plates denote replication, and points are model hyper-parameters. There are T image clusters and K segment clusters. The number of clusters is inferred from the data.

Each segment in an image has an associated descriptor, $\mathbf{x}_{jin} \in \mathbb{R}^{D_\mathbf{x}}$. These are distributed according to a mixture of K Gaussian clusters, which represent object-types,

$$\mathbf{x}_{jin} \sim \sum_{k=1}^{K} \beta_{tk} \mathcal{N}\big(\mathbf{x}_{jin} | \boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k^{-1}\big) . \tag{1}$$

Here $\beta_{tk} \in [0, 1]$ are the mixture weights, $\sum_k \beta_{tk} = 1$. The set of these k mixture weights, $\boldsymbol{\beta}_t$, is specific to each scene cluster or type, t. As is typical in Gaussian mixture models, we have an indicator variable $z_{jin}|(y_{ji} = t) \sim \text{Categ}(\boldsymbol{\beta}_t)$, which assigns segment observations to object-types (segment clusters) by taking a value in $\{1, \ldots, K\}$. Similarly, $y_{ji} \in \{1, \ldots, T\}$ is an indicator that assigns all segment indicators in an image, $\{z_{jin}\}_{n=1}^{N_{ji}}$, a distribution of objects, $\boldsymbol{\beta}_t$, to be drawn from. This allows each scene-type, t, to be composed of its own unique distribution of objects. We also put priors on the parameters so the number of clusters, K, can be inferred; $\boldsymbol{\beta}_t \sim \text{Dir}(\theta)$, $\boldsymbol{\mu}_k \sim \mathcal{N}\big(\mathbf{m}, (\gamma\boldsymbol{\Lambda}_k)^{-1}\big)$, and $\boldsymbol{\Lambda}_k \sim \mathcal{W}(\boldsymbol{\Omega}, \rho)$.

Each image also has an associated descriptor, $\mathbf{w}_{ji} \in \mathbb{R}^{D_\mathbf{w}}$, which is distributed according to an album-specific mixture of Gaussians;

$$\mathbf{w}_{ji} \sim \sum_{t=1}^{T} \pi_{jt} \mathcal{N}\big(\mathbf{w}_{ji} | \boldsymbol{\eta}_t, \boldsymbol{\Psi}_t^{-1}\big) . \tag{2}$$

Again $\pi_{jt} \in [0, 1]$ are mixture weights, $\sum_t \pi_{jt} = 1$. We re-use the indicators y_{ji} to assign each image descriptor to an associated image cluster, or scene-type, t. These indicators are drawn from a Categorical distribution, $y_{ji} \sim \text{Categ}(\boldsymbol{\pi}_j)$, where $\boldsymbol{\pi}_j$ is the set of all t mixture weights in album j. These weights are in turn drawn from a generalised Dirichlet distribution [6], $\boldsymbol{\pi}_j \sim \text{GDir}(a, b)$, which has a truncated stick-breaking representation [9],

$$\pi_{jt} = v_{jt} \prod_{s=1}^{t-1} (1 - v_{js}), \quad v_{jt} \sim \begin{cases} \text{Beta}(a, b) & \text{if } t < T \\ 1 & \text{if } t = T. \end{cases} \tag{3}$$

Here $v_{jt} \in [0, 1]$ are "stick-lengths" for each album. We have found empirically that using a generalised Dirichlet prior, as opposed to a Dirichlet prior, can help to limit the number of scene-types found, T, particularly for large datasets. Essentially, the generalised Dirichlet prior is being used as a simple, conjugate, parametric alternative to the HDP [22]. Like before, we also place priors on the Gaussian mixture parameters to help infer the number of scene types T; $\boldsymbol{\eta}_t \sim \mathcal{N}\big(\mathbf{h}, (\delta\boldsymbol{\Psi})^{-1}\big)$ and $\boldsymbol{\Psi}_t \sim \mathcal{W}(\boldsymbol{\Phi}, \xi)$.

A feature of the MCM is that scene-types are represented by a distribution of objects ($\boldsymbol{\beta}_t$) *as well as* a Gaussian scene-descriptor component. This allows the scene-descriptors to influence the type and co-occurrence of objects within a particular scene-type, potentially improving the object models. In this way the learned scene-type cluster is serving a similar role to a scene label as used by supervised algorithms. Also, the distributions of objects help to define the scene-type, potentially improving the scene-type clusters. All of this information is transferred through the shared y_{ji} indicator. The generative process for the MCM is;

1. Draw T image cluster parameters $\boldsymbol{\beta}_t$, $\boldsymbol{\eta}_t$ and $\boldsymbol{\Psi}_t$ from $\text{GDir}(a, b)$, $\mathcal{N}\big(\mathbf{h}, (\delta\boldsymbol{\Psi})^{-1}\big)$ and $\mathcal{W}(\boldsymbol{\Phi}, \xi)$ respectively.
2. Draw K segment cluster parameters $\boldsymbol{\mu}_k$ and $\boldsymbol{\Lambda}_k$ from $\mathcal{N}\big(\mathbf{m}, (\gamma\boldsymbol{\Lambda}_k)^{-1}\big)$ and $\mathcal{W}(\boldsymbol{\Omega}, \rho)$ 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 $\mathbf{x}_{jin} | (z_{jin} = k) \sim \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k)$.

Some readers may question the need for the whole-image descriptor, $\mathbf{w}_{ji}$, component of this model. We could just model scenes as "bags-of-segments" in a similar fashion to models presented in [7, 12]. We found such a model to consistently under-perform the MCM. This is because the image descriptors used by the MCM capture image spatial layout, which is typically absent in bag-of-segment/features approaches. We chose to model spatial layout at the descriptor level since it requires less computation complexity than modelling segment spatial layout directly in the MCM. This allows the MCM to scale to larger datasets, but does not allow spatial information to directly influence segment clustering. More details of the image and segment descriptors are given in section 4.

3. Variational Inference and Model Selection

To use Bayesian inference for learning the MCM's hyper-parameters, $\boldsymbol{\Xi} = \{a, b, \theta, \mathbf{h}, \delta, \boldsymbol{\Phi}, \xi, \mathbf{m}, \gamma, \boldsymbol{\Omega}, \rho\}$, we need to maximize the log-marginal-likelihood with respect to $\boldsymbol{\Xi}$,

$$\log p(\mathbf{W}, \mathbf{X} | \boldsymbol{\Xi}) = \log \int p(\mathbf{W}, \mathbf{X}, \mathbf{Y}, \mathbf{Z}, \boldsymbol{\Theta} | \boldsymbol{\Xi}) \, d\mathbf{Y} d\mathbf{Z} d\boldsymbol{\Theta}, \tag{4}$$

where $\boldsymbol{\Theta} = \{\boldsymbol{\pi}, \boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{\Lambda}, \boldsymbol{\eta}, \boldsymbol{\Psi}\}$ and the bold upper case letters represent the set of all of the respective lower case random variables. This integral is intractable, but a tractable lower bound can be found for the log-marginal-likelihood using variational Bayes. It is called free energy, $\mathcal{F}$, and the derivation of this lower bound is straight-forwardly computed using the method presented in [1]. Maximising $\mathcal{F}$ leads to the following approximating variational posterior distributions, $q(\cdot)$, over the image labels y_{ji},

$$\begin{aligned} q(y_{ji} = t) = \frac{1}{\mathcal{Z}_{y_{ji}}} \exp \Big\{ &\mathbb{E}_{q_\pi}[\log \pi_{jt}] \\ &+ \sum_{k=1}^{K} \mathbb{E}_{q_\beta}[\log \beta_{tk}] \sum_{n=1}^{N_{ji}} q(z_{jin} = k) \\ &+ \mathbb{E}_{q_{\boldsymbol{\eta},\boldsymbol{\Psi}}} \big[\log \mathcal{N}\big(\mathbf{w}_{ji} | \boldsymbol{\eta}_t, \boldsymbol{\Psi}_t^{-1}\big)\big] \Big\}. \end{aligned} \tag{5}$$

Here $\mathcal{Z}_{y_{ji}}$ is a normalisation constant and $\mathbb{E}_q[\cdot]$ are expectations with respect to the variational posterior distributions, all of which are given in the supplementary material. We can see in (5) that the image labels, y_{ji} are calculated as an exponential sum of Gaussian and Multinomial log-likelihoods, weighted by the jth album's mixture weights. The multinomial here (second term) is based on the number of segments belonging to object-type k in the jith image. Similarly, the variational posterior over the segment labels, z_{jin}, is,

$$\begin{aligned} q(z_{jin} = k) = \frac{1}{\mathcal{Z}_{z_{jin}}} \exp \Big\{ &\sum_{t=1}^{T} q(y_{ij} = t) \, \mathbb{E}_{q_\beta}[\log \beta_{tk}] \\ &+ \mathbb{E}_{q_{\boldsymbol{\mu},\boldsymbol{\Lambda}}} \big[\log \mathcal{N}\big(\mathbf{x}_{jin} | \boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k^{-1}\big)\big] \Big\}. \end{aligned} \tag{6}$$

Here the sum over the weighted image label probabilities, $q(y_{jin} = t)$, assigns more or less likelihood to the current segment cluster, k, based on the probability of the image belonging to a scene-type, t. Through the interaction of the labels in (5) and (6), the MCM transfers contextual information between "scene" and "object" co-occurrence.

Maximising $\mathcal{F}$ also leads to the following posterior hyper-parameters for the mixture weights;

$$\tilde{a}_{jt} = a + \sum_{i=1}^{I_j} q(y_{ji} = t)\,, \tag{7}$$

$$\tilde{b}_{jt} = b + \sum_{i=1}^{I_j} \sum_{s=t+1}^{T} q(y_{ji} = s)\,, \tag{8}$$

$$\tilde{\theta}_{tk} = \theta + \sum_{j=1}^{J} \sum_{i=1}^{I_j} q(y_{ji} = t) \sum_{n=1}^{N_{ji}} q(z_{jin} = k)\,. \tag{9}$$

These updates are essentially just the prior with added observation counts, or sufficient statistics. The sum in (8) must be performed in descending cluster size order, as per [10]. The MCM has similar updates for the Gaussian-Wishart hyper-parameters (in the supp. material). These indicator and hyper-parameter updates are iterated until $\mathcal{F}$ converges.

We use the following prior values for the hyper-parameters; $\theta, a, b, \gamma, \delta = 1$, $\rho = D_{\mathbf{x}}$, $\xi = D_{\mathbf{w}}$, $\mathbf{m} = \text{mean}(\mathbf{X})$, $\mathbf{h} = \text{mean}(\mathbf{W})$, $\boldsymbol{\Omega} = (\rho C_{width,s})^{\text{-1}}\mathbf{I}_{D_{\mathbf{x}}}$, and $\boldsymbol{\Phi} = (\xi \lambda^{\text{max}}_{\text{cov}(\mathbf{W})} C_{width,i})^{\text{-1}}\mathbf{I}_{D_{\mathbf{w}}}$. Here $\lambda^{\text{max}}_{\text{cov}(\mathbf{W})}$ 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 4. $C_{width,i}$ and $C_{width,s}$ (i for image, s for segment) are tunable parameters that encode the a-priori "width" of the mixtures (diagonal magnitude of the prior cluster covariances), and influence the number of clusters found. The rest of the hyperparameter values have been chosen for simplicity, and do not impact performance of the MCM greatly compared to the C_{width} parameters.

If the number of clusters, T and K, is known or set to some large value, the indicator and posterior hyper-parameter updates can 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 [10]. The MCM starts with $K = 1$ segment clusters, and iterates until convergence. Then the segment cluster is split in a direction perpendicular to its principal axis. These two clusters are then refined by running variational Bayes over them for a limited number of iterations. $\mathcal{F}$ 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 MCM.

In our "split-tally" heuristic, we greedily guess which cluster to split first by ranking all clusters' approximate contribution to $\mathcal{F}$ (details in the supplementary material). 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 much impact to 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, however we found that it was better to randomly initialise it to some large value, $T_{trunc} > T$, since both heuristics would interact.

4. Image Representation

Being an unsupervised algorithm, the MCM relies heavily on highly discriminative visual descriptors. We have chosen unsupervised feature learning algorithms for this task. They keep with the unsupervised theme of this work, and have lead to excellent performance in a number of classification tasks, e.g. [27].

4.1. Images $\mathbf{w}_{ji}$

For the image descriptors, $\mathbf{w}_{ji}$, we use a modified sparse coding spatial pyramid matching (ScSPM) descriptor [27]. For all experiments we use the original 1024-base Caltech-101 dictionary supplied by [27] to encode dense SIFT patches (16×16 pixels, with a stride of 8). We have found little to no reduction in classification and clustering performance doing this. Also, we use orthogonal matching pursuit (OMP) with 10 activations in place of the original sparse coding for large datasets since it is much less computationally demanding, and does not affect the MCM's 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 model, but we have found these codes are highly compressible with (randomised) PCA – to the point that we can compress them to $D_{\mathbf{w}} = 20$ while still achieving excellent image clustering performance.

4.2. Segments $\mathbf{x}_{jin}$

Out of the many segment descriptors tried, it was found that pooling dense independent component analysis (ICA) codes within segments gave the best results.

Firstly, we learn an under-complete ICA dictionary and its pseudo-inverse, $\mathbf{D}$ and $\mathbf{D}^+$, from at least 50,000 random image patches for each dataset. DC component removal and contrast normalising these patches was not necessary for well illuminated images. We then obtain ICA responses, $\mathbf{r}_l$, by multiplying patches with $\mathbf{D}^+$, centred on *every* pixel, $l \in [1, L]$, in every image. L is the total number of pixels in an image. Then the fast mean shift algorithm [5] is used to segment the original images into sets of pixels, S_{jin}. Typically we obtain less than 20 segments per image. The ICA responses are transformed and mean-pooled within each segment in the following manner:

$$\mathbf{x}'_{jin} = \frac{1}{\# S_{jin}} \sum_{l \in S_{jin}} \log |\mathbf{r}_l| \tag{10}$$

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

Both of these descriptors take about 1 second each per image to calculate. The ScSPM and ICA features are complementary. 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. The structure of the MCM works well with these complementary representations.

5. Experiments

In this section we compare the MCM to other algorithms in the literature. We use three standard datasets (single album) and a large novel dataset consisting of twelve surveys (albums) from an autonomous underwater vehicle (AUV).

Normalised mutual information (NMI) [20] 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 and weakly-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. Hence this measure must be viewed with caution. NMI does not suffer from these problems. We do not use training or test sets since no labels are used by the algorithm. Some of the algorithms we compare against use different splits of training and testing data. Unfortunately there is no closed form solution for predicting labels on new data with the MCM.

We also compare the MCM to the unsupervised variational Dirichlet process (VDP) of [10], latent Dirichlet allocation [3] with Gaussian clusters (G-LDA), and self-tuning spectral clustering (SC) [28]. The VDP, G-LDA, and SC can only cluster one descriptor source (image *or* segment) at once. We have modified the VDP and G-LDA to use the same cluster splitting heuristic discussed in section 3, and the same priors as the MCM. They are entirely deterministic algorithms, since they do not require a random initialisation like the MCM. We use SC for clustering the image ScSPM descriptors only, and we run it with 10 random starts of K-means given the true number of classes.

For all datasets, the MCM has a truncation level $T_{trunc} = 30$, and is run ten times with a random initialisation of $\mathbf{Y}$. The VDP, G-LDA 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 Core 2 Duo 3 GHz system with 6GB RAM unless otherwise stated.

5.1. Microsoft Research Classes

The first dataset considered is Microsoft's MSRC v2 dataset, which has both scene and object labels and is used by [7, 12]. We use the same 10 scene categories as these papers, 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 weakly- and supervised algorithms. The VDP+ScSPM refers to running the VDP with the image ScSPM based descriptors. The MCM performs substantially better than the VDP and SC for this dataset (which struggle with so few im-

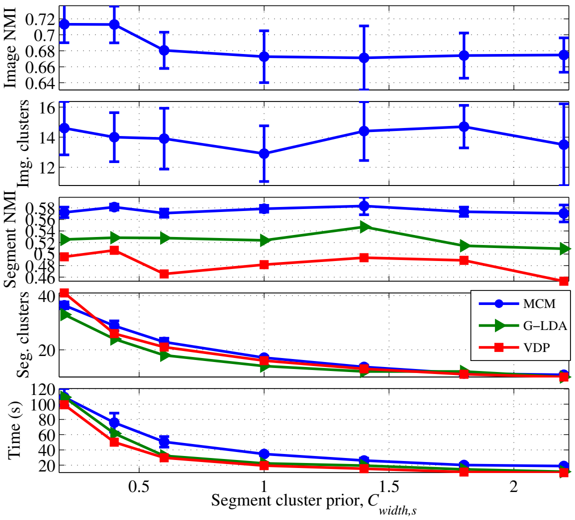


Figure 3. Segment performance on the MSRC v2 dataset. The MCM uses $C_{width,i} = 0.08$ for the image cluster prior. The VDP and G-LDA can only cluster $\mathbf{x}_{jin}$ here.

Table 1. Image performance for the MSRC dataset. The MCM uses $C_{width,i} = 0.08$ and $C_{width,s} = 0.4$. The VDP uses $C_{width,i} = 0.02$ and finds $T = 14$. More statistics for the MCM are shown in Figure 3. #0 indicates the average number of unassigned classes, or zeros, on the diagonal of the confusion matrix.

Algorithm	**NMI** (std.)	**Acc.** (% (std.), #0)
MCM	0.713 (0.023)	72.0 (3.3), 1.1
VDP+ScSPM [10]	0.636	56.69, 2
SC+ScSPM [28]	0.643 (0.002)	66.1 (1.6), 2.1
L^2-LEM-χ^2 [24]	0.554 (0.018)	62.0 (2.7), 1.1
Du *et. al.* [7]	0.745	82.9
Du *et. al.* [7] LSBP	0.801	86.8
Li *et. al.* [12]	**0.820**	**89.06**

ages), but does lag behind the weakly-supervised methods of [7, 12]. However, the MCM still manages to achieve visually consistent image and segment clusters, see Figure 1.

Object 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. 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 MCM's segment clustering performance we ran it for an array of $C_{width,s}$ values and compared it against the unsupervised VDP and G-LDA algorithms. The VDP clusters all segments without any notion of context, and G-LDA can model each image as having its own proportions of segment clusters (image context). The results are summarised in Figure 3. We can see that the MCM, which models scene-type context, consistently outperforms the VDP and G-LDA.

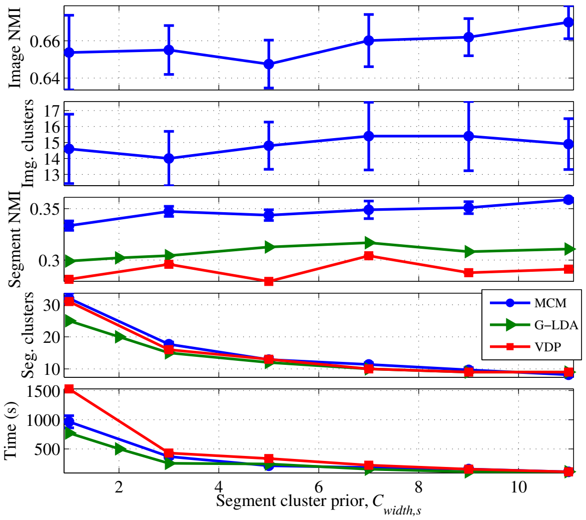


Figure 4. Segment performance for LabelMe, $C_{width,i} = 0.08$.

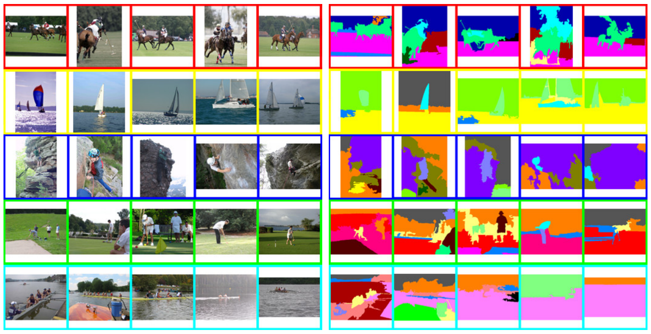
Table 2. Image performance for the LabelMe dataset for $C_{width,i} = 0.08$ (both MCM and VDP), and $C_{width,s} = 11$. The VDP finds $T = 8$.

Algorithm	**NMI** (std.)	**Acc.** (% (std.), #0)
MCM	0.670 (0.009)	80.0 (2.8), 0.1
VDP+ScSPM [10]	0.708	82.3, 0
SC+ScSPM [28]	0.679 (0.017)	74.1 (3.5), 1.1
Li *et. al.* [12]	0.600	76.25
sLDA [25]	0.606	76
sLDA [25] (annots.)	0.606	76
DiscLDA+GC [15]	0.646	81
SVM + ScSPM [27]	0.6958	84.38
CA-TM [15]	**0.729**	**87**

5.2. LabelMe

The next dataset we used was obtained from LabelMe [18], and has been used by [12, 25, 15]. It is comprised of 2688 images (256 × 256 pixels), with 8 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 (given in the supplementary material). The appearance of these object classes is far less constrained than the other datasets.

Again we compare the MCM to state-of-the-art methods in Table 2. The MCM is quite competitive on this dataset. Interestingly, it does not perform as well as the VDP using the modified ScSPM descriptors, which even outperforms an SVM using unmodified ScSPM descriptors [27]. The



(a) Image clusters.

(b) Segment clusters.

Figure 5. A random selection of images from 5 of the 10 image clusters found by the MCM on the UIUC dataset (a), with some of the (30) corresponding segment clusters in (b). The image clusters have a NMI score of 0.652, and an estimated accuracy of 74.0%.

Table 3. Image performance for UIUC sport events, $C_{width,i} = 0.16$ and $C_{width,s} = 1$, with mean $T = 11.3$, $K = 30.2$, and runtime 444.61s. The VDP uses $C_{width,i} = 0.12$ and finds T = 6.

Algorithm	**NMI** (std.)	**Acc.** (% (std.), #0)
MCM	**0.641** (0.018)	74.1 (1.5), 1
VDP+ScSPM [10]	0.557	63.4, 2
SC+ScSPM [28]	0.429 (0.02)	58.9 (2.4), 1.1
Du *et. al.* [7] no LSBP	0.389	60.5
Du *et. al.* [7] LSBP	0.418	63.5
Li *et. al.* [13]	0.276	54
sLDA [25] (annots.)	0.438	66
sLDA [25]	0.446	65
Li *et. al.* [12]	0.466	69.11
DiscLDA+GC [15]	0.506	70
SVM+ScSPM [27]	0.549	72.9
CA-TM [15]	0.592	**78**

MCM also appears to perform slightly worse than SC, but they are within one standard deviation. In this case it seems the segment clusters may be confounding the MCM image clustering somewhat.

From Figure 4 we can see the MCM again far outperforms the other unsupervised algorithms for segment clustering, demonstrating the importance of scene-type context for object recognition.

5.3. UIUC Sport Events

The final standard dataset is the UIUC sports dataset used by [7, 12, 13, 15, 25]. This dataset depicts 8 types of sporting events and has 1579 images (maximum dimension of 320 pixels), unfortunately it has no segment labels. We use the same segment descriptor settings as the MSRC dataset. Results for image clustering/classification are presented in Table 3. Note that the algorithm from [7] 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. Somewhat surprisingly the MCM is one of the best performing algorithms on this dataset. An example MCM result is shown in Figure 5.

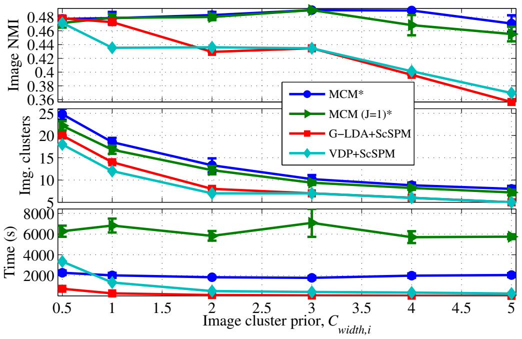


Figure 6. Image performance AUV dataset, $C_{width,s} = 350$, 100,647 images. * Denotes 8 Xeon 2.2 GHz cores were used.

5.4. Robotic 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 [26]. This datasets has 100,647 downward looking stereo pair images taken from an altitude of 2 m. 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 before descriptor extraction. 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, though many of these classes are quite visually similar so the labels have a small amount of noise.

All 100,647 images were clustered with the MCM, VDP and G-LDA (at the image level) while varying $C_{width,i}$, see Figure 6. Both the MCM and G-LDA can model the 12 separate surveys as individual groups, j. We also clustered this dataset using the MCM with the surveys concatenated into one group ($J = 1$). In this way we can quantitatively determine the utility of modelling groups. This is also achieved by comparing G-LDA and VDP, the latter does not model groups. The MCM variants are run with 8 Xeon (E5-4260) 2.2 GHz cores, unlike the VDP and G-LDA. This is done to demonstrate these algorithms are parallelisable, and to expedite the running of these experiments (the MCM variants have to cluster 1.7 million segment descriptors).

In Figure 6 we can see that the MCM variants show quite consistent NMI performance throughout the range of

$C_{width,i}$ values. NMI for G-LDA and the VDP drop off quite quickly for increasing $C_{width,i}$. There is also not a huge difference in NMI between the $J = 1$ and $J = 12$ models. However, G-LDA consistently has a faster runtime than the VDP despite no parallelism. Similarly, the MCM with groups has a much faster run time than the MCM with $J = 1$. This can be partially attributed to the way the MCM is parallelised, but not to the extent observed. We conjecture that modelling groups helps to separate the latent clusters in the data since clusters may not co-occur in all groups.

6. Conclusion

This paper has demonstrated that fully unsupervised, annotation-less algorithms for scene understanding can be competitive with supervised and weakly-supervised algorithms. The proposed MCM can use contextual information from scene-types to improve object discovery, and in three of the four experiments, is able to use object co-occurrence and proportion information to greatly improve scene discovery. We have also demonstrated that the MCM is able to run on large datasets gathered by autonomous robots, enabling fully automated data gathering and interpretation pipelines. Like many weakly- and supervised scene understanding models, the MCM is effective at discovering scene-types, but not as effective at object discovery – which is a much harder problem. Focusing on the unsupervised object discovery and recognition aspects of such models will be a useful area of future research. The MCM can form useful representations of visual data without incorporating any semantic knowledge, so it may form a good basis for models that are robust to label noise at both the image and objects levels. Such models would also be useful in active learning scenarios for labelling, interpretation and analysis of scientific datasets.

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] M. J. Beal. *Variational algorithms for approximate Bayesian inference*. PhD thesis, University College London, 2003. 3

[2] D. M. Blei and 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, pages 127–134, New York, NY, USA, 2003. ACM. 1

[3] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet allocation. *The Journal of Machine Learning Research*, 3:993–1022, 2003. 5

[4] L. Cao and L. Fei-Fei. Spatially coherent latent topic model for concurrent segmentation and classification of objects and scenes. In *International Conference Computer Vision, ICCV*, pages 1–8. IEEE, 2007. 1, 2

[5] C. Christoudias, B. Georgescu, and P. Meer. Synergism in low level vision. In *16th International Conference on Pattern Recognition*, volume 4, pages 150–155 vol.4, 2002. 5

[6] R. J. Connor and J. E. Mosimann. Concepts of independence for proportions with a generalization of the Dirichlet distribution. *Journal of the American Statistical Association*, 64(325):194–206, 1969. 3

[7] L. Du, L. Ren, D. Dunson, and L. Carin. A Bayesian model for simultaneous image clustering, annotation and object segmentation. *Advances in Neural Information Processing Systems*, 22:486–494, 2009. 1, 3, 5, 6, 7

[8] R. Gomes, M. Welling, and P. Perona. Incremental learning of nonparametric Bayesian mixture models. In *Computer Vision and Pattern Recognition, CVPR.*, pages 1–8. IEEE, 2008. 2

[9] H. Ishwaran and L. F. James. Gibbs sampling methods for stick-breaking priors. *Journal of the American Statistical Association*, 96(453):161–173, 2001. 3

[10] K. Kurihara, M. Welling, and N. Vlassis. Accelerated variational Dirichlet process mixtures. *Advances in Neural Information Processing Systems*, 19:761, 2007. 4, 5, 6, 7

[11] Y. J. Lee and K. Grauman. Object-graphs for context-aware category discovery. In *Computer Vision and Pattern Recognition, CVPR.*, pages 1–8. IEEE, 2010. 2

[12] L. Li, M. Zhou, G. Sapiro, and L. Carin. On the integration of topic modeling and dictionary learning. *International Conference on Machine Learning, ICML*, 2011. 1, 3, 5, 6, 7

[13] L.-J. Li, R. Socher, and L. Fei-Fei. Towards total scene understanding: Classification, annotation and segmentation in an automatic framework. In *Computer Vision and Pattern Recognition, CVPR 2009*, pages 2036–2043. IEEE, 2009. 1, 7

[14] N. Loeff and A. Farhadi. Scene discovery by matrix factorization. In *European Conference on Computer Vision, ECCV*, pages 451–464. Springer, 2008. 2

[15] Z. Niu, G. Hua, X. Gao, and Q. Tian. Context aware topic model for scene recognition. In *Computer Vision and Pattern Recognition, CVPR*, pages 2743 –2750. IEEE, 2012. 1, 6, 7

[16] A. Oliva and A. Torralba. Building the gist of a scene: The role of global image features in recognition. *Progress in Brain Research*, 155:23, 2006. 2

[17] B. C. Russell, W. T. Freeman, A. A. Efros, J. Sivic, and A. Zisserman. Using multiple segmentations to discover objects and their extent in image collections. In *Computer Vision and Pattern Recognition, CVPR*, pages 1605–1614. IEEE, 2006. 2

[18] B. C. Russell, A. Torralba, K. Murphy, and W. Freeman. LabelMe: A database and web-based tool for image annotation. *International Journal of Computer Vision*, 77:157–173, 2008. 10.1007/s11263-007-0090-8. 6

[19] J. Sivic, B. C. Russell, A. Zisserman, W. T. Freeman, and A. A. Efros. Unsupervised discovery of visual object class hierarchies. In *Computer Vision and Pattern Recognition, CVPR*, pages 1–8. IEEE, 2008. 2

[20] A. Strehl and J. Ghosh. Cluster ensembles – a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, 3:583–617, 2003. 5

[21] E. Sudderth, A. Torralba, W. Freeman, and A. Willsky. Learning hierarchical models of scenes, objects, and parts. In *Internation Conference on Computer Vision, ICCV*, volume 2, pages 1331–1338. IEEE, 2005. 1

[22] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581, 2006. 3

[23] A. Torralba, K. P. Murphy, and W. T. Freeman. Using the forest to see the trees: exploiting context for visual object detection and localization. *Commun. ACM*, 53(3):107–114, 2010. 2

[24] T. Tuytelaars, C. H. Lampert, M. B. Blaschko, and W. Buntine. Unsupervised object discovery: A comparison. *International Journal of Computer Vision*, 88(2):284–302, 2010. 2, 6

[25] C. Wang, D. Blei, and L. Fei-Fei. Simultaneous image classification and annotation. In *Computer Vision and Pattern Recognition, CVPR*, pages 1903–1910. IEEE, 2009. 1, 6, 7

[26] 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. Steinberg, and A. Friedman. Monitoring of benthic reference sites: using an autonomous underwater vehicle. *Robotics Automation Magazine, IEEE*, 19(1):73–84, 2012. 7

[27] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. In *Computer Vision and Pattern Recognition, CVPR*, pages 1794–1801. IEEE, 2009. 4, 6, 7

[28] L. Zelnik-Manor and P. Perona. Self-tuning spectral clustering. In *Advances in Neural Information Processing Systems*, volume 17, pages 1601–1608, 2004. 5, 6, 7