An Unsupervised Approach to Modelling Visual Data

Daniel Matthew Steinberg

A thesis submitted in fulfillment
of the requirements of the degree of
Doctor of Philosophy

THE UNIVERSITY OF
SYDNEY

Australian Centre for Field Robotics
School of Aerospace, Mechanical and Mechatronic Engineering
The University of Sydney

July 2013

Declaration

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the University or other institute of higher learning, except where due acknowledgement has been made in the text.

Daniel Matthew Steinberg

July 2013

Abstract

Daniel Matthew Steinberg
The University of Sydney

Doctor of Philosophy
July 2013

An Unsupervised Approach to Modelling Visual Data

With the advent of cheap, high fidelity, digital imaging systems it is now easy to create huge collections of digital images. Subsequently, the computer vision community has seen an explosion of research in classifying these images into scenes, recognising objects within images, propagating user tags to new images, and even attempts at whole image "understanding".

Most of this research uses *supervised* or *semi-supervised* algorithms, which rely upon some form of human generated "ground-truth". For very large scientific datasets with many classes, producing the ground-truth data can represent a substantial, and potentially expensive, human effort. In these situations there is scope for the use of unsupervised approaches that can model collections of images and automatically summarise their content. The primary motivation for this thesis comes from the problem of labelling large visual datasets of the seafloor obtained by an autonomous underwater vehicle (AUV) for ecological analysis. It is expensive to label this data, as taxonomical experts for the specific region are required. Quick, approximate summaries of quasi-habitats and objects within images can be generated by unsupervised methods "for free". These can be used to focus the efforts of experts, and inform decisions on additional sampling. These techniques are equally applicable to large photo albums and collections, such as the millions of images hosted on sites like *Flickr*, where image annotations may be incorrect or absent entirely.

The contributions in this thesis arise from modelling this visual data in entirely unsupervised ways to obtain comprehensive visual summaries for subsequent expert annotation. Firstly, popular unsupervised image feature learning approaches are adapted to work with large datasets and unsupervised clustering algorithms. Next, using Bayesian models the performance of rudimentary scene clustering is boosted by sharing clusters between multiple related datasets, such as photo albums or AUV surveys. Then these Bayesian scene clustering models are extended to simultaneously cluster sub-image super-pixels, or segments, to form unsupervised notions of "objects" within

scenes. The frequency distribution of these objects within scenes is used as the scene descriptor ("bag-of-segments") for simultaneous scene clustering. This model also takes advantage of multiple related datasets, and its various properties are shown to enhance clustering through the use of contextual information inherent within the data. Finally, this simultaneous clustering model is extended to make use of whole image descriptors, which encode rudimentary spatial information, as well as object frequency distributions to describe scenes. This is achieved by unifying the previously presented Bayesian clustering models, and in so doing rectifies some of their weaknesses and limitations. Hence, the final contribution of this thesis is a practical *unsupervised* algorithm for modelling images from the super-pixel to album levels, and is applicable to large datasets.

Acknowledgements

Firstly I would like to thank my supervisor, A. Prof. **Stefan Williams** and co-supervisor, Dr. **Oscar Pizarro**. You have both been excellent mentors and friends during both my undergraduate and graduate studies. Thank you for the fantastic advice, opportunities and field trips you have provided over the years, and most of all, thank you for keeping the research we do fun!

Also thank you to the external reviewers of this thesis; **Mark Cummins**, **Bryan Russell** and **Gregory Dudek**. I very much appreciate the time you all put into reading this thesis, and for providing very insightful and detailed feedback.

My time at the Australian Centre for Field Robotics (ACFR) has been life-changing, and a large part of this is because of the excellent friends I have made here. Ash Bender and Ariell Friedman (whom I have known since primary school and undergrad respectively), and John Vial, Lachlan McCalman, Alastair Quadros, Peter Morton, Mark De Deuge, Don Dansereau, Lachlan Toohey, Dan Bongiorno and Michael Bewley. Thanks for all the commiseration over excellent coffee, talking about geeky stuff at the pub, and the really fun camping trips and holidays! It's very sad this is all coming to the end for many of us, and I will really miss you all when we all eventually go our separate ways.

I would also like to thank the rest of the members of the marine robotics group, past and present, including but not limited to post-docs Mike Jakuba, Bart Douillard, Matt Johnson-Roberson, Mitch Bryson, Ian Mahon, and Navid Nourani-Vatani. Also Ritesh Lal and the rest of the technical staff for all of the advice, feedback, and keeping things running smoothly. Also thank you to all of the academics and staff of the ACFR for creating such a wonderful working and learning environment.

Thank you, and much love to my family. Thanks Mum and Dad for encouraging me, and supporting me all through my life. Without your support and encouragement I doubt I would have ever undertaken this study, and I would have missed a huge opportunity. Thank you to my sisters for all the hysterical times and distractions from reality. Also thank you to my grandmother and late grandfather, who both played a large part in setting me on this path in life. Finally, thank you to Amanda for always being there and making each day fun, thanks for lending a sympathetic ear, and helping me tune out and just enjoy life!

To all of the fabulous friends I have made while researching for this thesis – you have really made the time fly...

Contents

Declaration i

Abstract iii

Acknowledgements v

1 Introduction 1

1.1 Background and Motivation 1

1.2 Problem Statement 4

1.3 Contributions 6

1.4 Outline 7

2 Background 11

2.1 An Overview of some Models for Unsupervised Learning 11

2.1.1 Clustering 12

2.1.2 Topic Models 19

2.2 Primer on Variational Bayes 23

2.2.1 Derivation of Variational Bayes 23

2.2.2 Application to a Conjugate Exponential Mixture Model 27

2.3 Modelling Visual Data – Literature Review 34
2.3.1 Unsupervised Scene Recognition 35
2.3.2 Supervised Scene Recognition 37
2.3.3 Object Recognition and Discovery 38
2.3.4 Scene Understanding 40

3 Adapting Feature Learning for Large Scale Image Clustering 43

3.1 Introduction 44
3.2 Sparse Coding Overview 46
3.3 Image Coding Framework 48
3.4 Dimensionality Reduction for Image Descriptors 51
3.5 Experiments 53
3.5.1 Comparing Sparse Encoders 56
3.5.2 Number of Dimensions to Preserve 58
3.5.3 Dictionary Comparison 62
3.6 Summary 69

4 Clustering Groups of Related Visual Datasets 73

4.1 Introduction 74
4.2 Sharing Clusters Between Groups 77
4.3 Variational Bayes for learning the Model 80
4.4 Model Variants 83
4.5 Image Representation 84
4.6 Experiments 85

4.6.1 Number and Composition of Albums/Groups 87
4.6.2 Effect of Cluster Hyper Parameter Values 91
4.6.3 Case Study on a Scientific Dataset 93
4.6.4 Case Study on a Photo Collection 97
4.7 Summary 100

5 Clustering Multiple Levels of Related Visual Datasets 105

5.1 Introduction 106
5.2 Clustering Multiple Levels of Images Over Multiple Datasets 109
5.3 Variational Bayes for Learning the Models 112
5.4 Model Variants 117
5.5 Image Representation 117
5.6 Experiments 120
5.6.1 Contextual Effects on Image Clustering 122
5.6.2 Number of Albums/Groups 128
5.6.3 Model Prior 129
5.6.4 Case study on a Scientific Dataset 131
5.6.5 Case Study on a Photo Collection 137
5.7 Summary 138

6 Clustering Observations of Images and Image Parts 145

6.1 Introduction 146
6.2 Clustering Observations of Images and Image Parts in Groups 147
6.3 Variational Bayes for Learning the Model 150

6.4 Image Representation 152
6.5 Experiments 153
6.5.1 Effects of Clustering Observations of Images and Image Parts 155
6.5.2 Case Study on a Scientific Dataset 159
6.5.3 Case Study on a Photo Collection 163
6.6 Summary 165

7 Conclusion **167**
7.1 Summary of Contributions 167
7.1.1 Large Scale Adaptation and Analysis of Sparse Coding Spatial Pyramids 168
7.1.2 Clustering Multiple Related Datasets Jointly 169
7.1.3 An Analysis of Context and Simultaneous Clustering 170
7.1.4 Combining Models for a Richer Image Representation 171
7.2 Future Work 171
7.2.1 Integrating Sparse Coding, Pooling and Dimensionality Reduction 171
7.2.2 Clarifying the Relationship Between Groups, Classes, and Distributions 172
7.2.3 Exploiting more Context within Images 173
7.2.4 Extensions to Semi-Supervised Learning 174

Bibliography **175**

A Some Useful Distributions and Expectations for Variational Bayes **187**
A.1 Exponential Family 187

A.1.1 Expectations over the likelihood 188
A.1.2 Variational updates 188
A.1.3 Free energy expectations 188
A.2 Dirichlet Distribution 188
A.2.1 Expectations over the likelihood 189
A.2.2 Variational updates 189
A.2.3 Free energy expectations 190
A.3 Generalised Dirichlet Distribution 190
A.3.1 Expectations over the likelihood 190
A.3.2 Variational updates 191
A.3.3 Free energy expectations 191
A.4 Gaussian-Wishart Distribution 192
A.4.1 Expectations over the likelihood 192
A.4.2 Variational updates 193
A.4.3 Free energy expectations 193

B Functional Derivatives **195**

List of Figures

1.1 Summary of recent AUV campaigns 2

1.2 Automated data collection and summary 4

1.3 Objects in context 5

2.1 Graphical models of a Gaussian mixture model (GMM) and Bayesian Gaussian mixture model (BGMM) 14

2.2 Graphical model of smoothed latent Dirichlet allocation (LDA) 22

2.3 Graphical model of a Bayesian mixture model 29

2.4 Sample results of clustering imagery from an AUV 35

3.1 The sparse code spatial pyramid matching (ScSPM) pipeline 49

3.2 Exemplar images of the three datasets used for comparison 55

3.3 First two principal axes of encoder feature spaces on the outdoor scenes dataset 60

3.4 Linear support vector machine (SVM) and K-means performance on compressed sparse codes 61

3.5 The top 100 Eigenvalues from principal component analysis (PCA) on all datasets 62

3.6 SVM classification using PCA compressed codes on all of Caltech-101 63

3.7 Eigenvalues of the original scale-invariant feature transform (SIFT) descriptors 67

3.8 Evaluation of dictionary size and training set on classification and clustering performance 68

4.1 Demonstration of jointly clustering datasets (groups) 75

4.2 Graphical model of the grouped mixtures clustering model (GMC) and a Bayesian mixture model 79

4.3 Example artificial groups 88

4.4 The effect of the type, and number of groups on clustering 90

4.5 Example clustering from outdoor scenes dataset 91

4.6 The effect of prior cluster width on clustering in groups 92

4.7 Group clustering results on the AUV dataset 95

4.8 Cluster examples from, and overview of the AUV dataset 96

4.9 C_{width} effects on held-out log-likelihood, $\hat{\mathcal{L}}$, for the AUV dataset. . . . 97

4.10 Group clustering results on the holiday albums dataset 98

4.11 Ranked examples of GMC clusters for the holidays dataset 99

4.12 Ranked examples of variational Dirichlet process (VDP) clusters for the holidays dataset 100

4.13 Random examples of GMC clusters for the holidays dataset 101

4.14 Random examples of VDP clusters for the holidays dataset 102

5.1 Demonstration of simultaneous clustering in groups 107

5.2 Simultaneous Clustering Models 111

5.3 Independent component analysis (ICA) based super-pixel descriptors. 119

5.4 Effect of context on clustering results 124
5.5 Sample MSRC-2 simultaneous clustering result 125
5.6 Sample MSRC-2 segment clusters 126
5.7 Sample outdoor scene simultaneous clustering result 127
5.8 Context and segment homogeneity and completeness 128
5.9 Effect of number of groups on clustering results 129
5.10 Effect of priors on clustering results 131
5.11 Context and image homogeneity and completeness 132
5.12 Some exemplar images of the AUV dataset ground truth classes. . . . 132
5.13 Sample simultaneous clustering results on the AUV dataset 134
5.14 Sample AUV segment clusters 135
5.15 How the simultaneous clustering model (SCM) image cluster prior, ϕ, affects clustering 136
5.16 Sample SCM clustering results on the photo album dataset 139

6.1 Demonstration of clustering multiple observation sources in groups . . 146
6.2 The multiple-source clustering model (MCM) graphical model 150
6.3 Effect of context on clustering results 156
6.4 Sample MSRC-2 MCM clustering result 157
6.5 Sample MSRC-2 segment clusters 158
6.6 Comparing the MCM and BGMM for image clustering 158
6.7 The MCM on the AUV dataset 160
6.8 MCM sample cluster results on the AUV dataset 161
6.9 Sample AUV segment clusters 162

6.10 Image and segment cluster distributions per group 162

6.11 Comparing the MCM, GMC and BGMM for image clustering on the AUV dataset 163

6.12 Sample MCM clustering results on the photo album dataset 164

List of Tables

3.1 SVM results on $\mathbb{R}^D$ features 57

3.2 PCA+K-means results on $\mathbb{R}^d$ features 58

3.3 Random+K-means results on $\mathbb{R}^d$ features 59

3.4 Orthogonal matching pursuit (OMP)+SVM cross-dataset dictionary comparison 64

3.5 Sparse coding (SC)+SVM cross-dataset dictionary comparison 64

3.6 OMP+K-means cross-dataset dictionary comparison 66

3.7 SC+K-means cross-dataset dictionary comparison 66

4.1 Summary of non-group clustering (and classification) models for the outdoor scenes dataset. 89

5.1 Comparison of clustering algorithms for images 143

5.2 AUV dataset label summary 144

List of Algorithms

2.1 The Bayesian exponential mixture model variational Bayes (VB) algorithm 34

4.1 The GMC exhaustive model selection heuristic 83

5.1 The SCM greedy model selection heuristic 143

Notes on Notation

All lower-case bold symbols and letters denote *column* vectors unless otherwise stated. Bold upper-case symbols and letters can either denote sets or matrices, and their exact definition is made explicit in the text, with sets more commonly used (though this distinction in most cases may not matter). Square brackets $[\ldots]$ denote vector or matrix construction through concatenation of scalars or vectors respectively. Curly braces $\{\ldots\}$ denote set construction.

The following conventions are used for representing probability distributions. A probability distribution on $\mathbf{x}$ with parameters θ is denoted $p(\mathbf{x}|\theta)$. When the generative process is mentioned, such as "$\mathbf{x}$ is drawn from a distribution with parameters θ", the convention $\mathbf{x} \sim p(\theta)$ is adopted. The same distribution is referred to in both these instances. However, when $p(\theta)$ is mentioned on its own, it refers to the distribution over θ.

Chapter 1

Introduction

1.1 Background and Motivation

With the advent of cheap, high fidelity digital imaging systems, it is now easy for anyone to create huge collections of digital images. Managing these ever-growing collections of images now far exceeds the patience, if not also the capabilities, of people. Internet sites like *Flickr* and *Picasa web albums* etc. now allow for the hosting of these collections online, with the option of adding textual tags and annotations to these images. Consequently, the computer vision community has seen an explosion of research focused on classifying this imagery into scenes [41, 84], identifying objects [30, 78], to more holistic supervised and semi-supervised "image understanding" [39, 72, 73]. This is a fascinating and highly active research area. However, most of this research is focused on *supervised*, or *semi-supervised* learning. That is, there has to be some "ground-truth" training data on which to train, or inform, these algorithms, which is ubiquitously provided by humans. In most cases this training data is relatively easy to obtain, especially from the aforementioned websites. Though, this training data is often not of the highest quality and dealing with this is in itself an open research question [72].

Not all visual data can be labelled or annotated easily. An example of this is data used for scientific research, which needs to be annotated by an expert in the field. This is

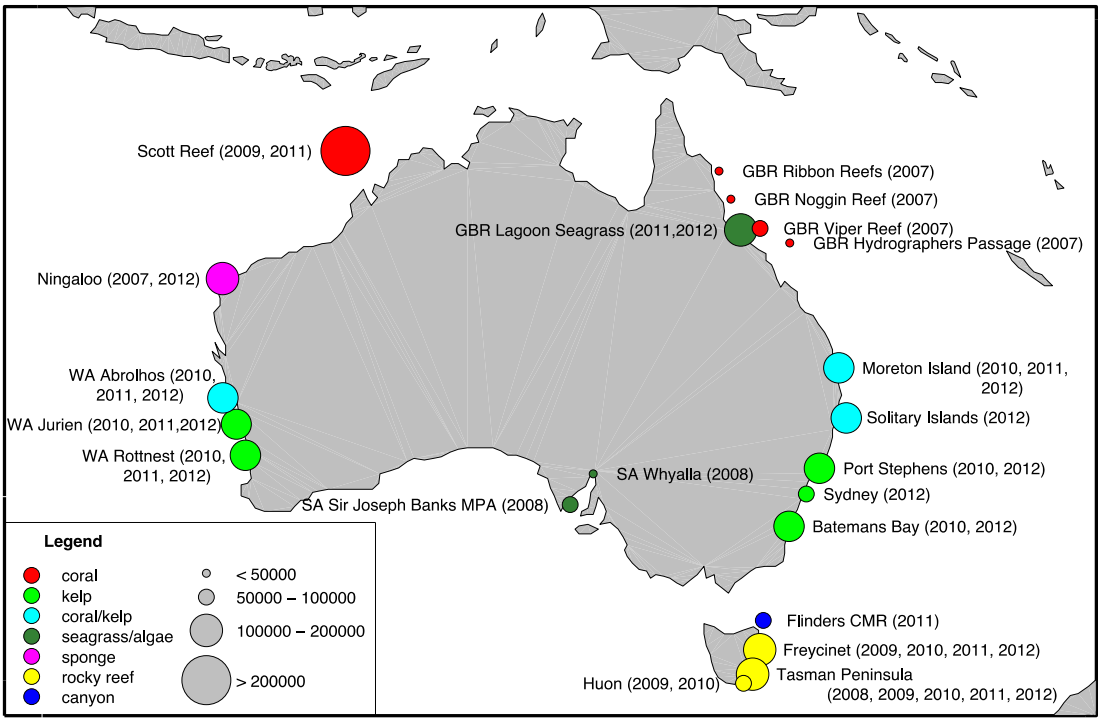


Figure 1.1 – autonomous underwater vehicle (AUV) campaigns undertaken by the marine robotics group at the Australian Centre for Field Robotics (ACFR) nationally. The size of the circles indicates approximately how many stereo pairs were collected, and the colour corresponds to the dominant habitat type.

of paramount importance since scientific conclusions, which may have wide reaching impacts, are determined by this data. Again, the advent of inexpensive digital imaging systems have enabled the acquisition of truly massive scientific datasets. For example, underwater ship-towed video systems used to characterise the benthos[1] collect hours of videos in a single deployment. In some cases this can add up to hundreds of gigabytes of video from a single field trip. And quite often this will only be part of a smaller multi-year study to, for example, detect environmental change. Ideally every frame of this video needs to be analysed and annotated for habitat type, and flora and fauna present.

Similarly, the Australian Centre for Field Robotics (ACFR) marine robotics group provides autonomous underwater vehicle (AUV) infrastructure to many marine scien-

[1]Organisms that live on, or near the seabed.

tists in Australia as part of a national collaborative research infrastructure scheme[2]. The AUVs used have downward pointing stereo cameras, and are used to create accurate 3D models of the benthos [62]. Each time a vehicle is deployed, tens of thousands of high quality images are collected of the benthos, and there may be in excess of ten AUV deployments per field trip. A summary of recent AUV campaigns is presented in Figure 1.1, courtesy of [125]. Again, labelling this imagery requires experts in taxonomy local to the region. Financial remuneration for these experts can cut into the budgeting of already expensive research. Consequently, much of this data goes unused as only small subsets of the data can be expertly analysed.

Another example where it is hard to obtain human labels is in the interpretation of imagery on an extra-planetary rover in novel terrain, where communication may have significant bandwidth limitations and delays. It may be of use to identify and only transmit a summary that most compactly represents what is seen, as in Thompson et al. [112]. Of course, this reasoning could also be applied to the keen photographer who is also a lazy personal photo album manager.

The difficulty, and expense, of labelling even a small amount of these large scientific visual datasets is the primary motivation of this thesis. This is an area where *unsupervised* data exploration techniques, which do not require training or labelled data, can have a large impact on scientific outcomes. In the most simple case, clustering imagery into groups that have similar appearance based on texture, colour and structure, can provide useful visual summaries of the data. These clustering methods also provide coarse labels, which in conjunction with the summaries, can be used to drastically simplify, and focus, subsequent annotation efforts by experts. An example of how this clustering can be used, in conjunction with 3D reconstructions of the benthos, is presented in Figure 1.2. Some of these clustering models [63, 103], which are precursors to those presented in this thesis, have already been used to aid marine ecologists in focusing their studies and annotation efforts to only relevant imagery that contain the biota of interest [28].

[2]Integrated Marine Observing System (IMOS), http://www.imos.org.au/.

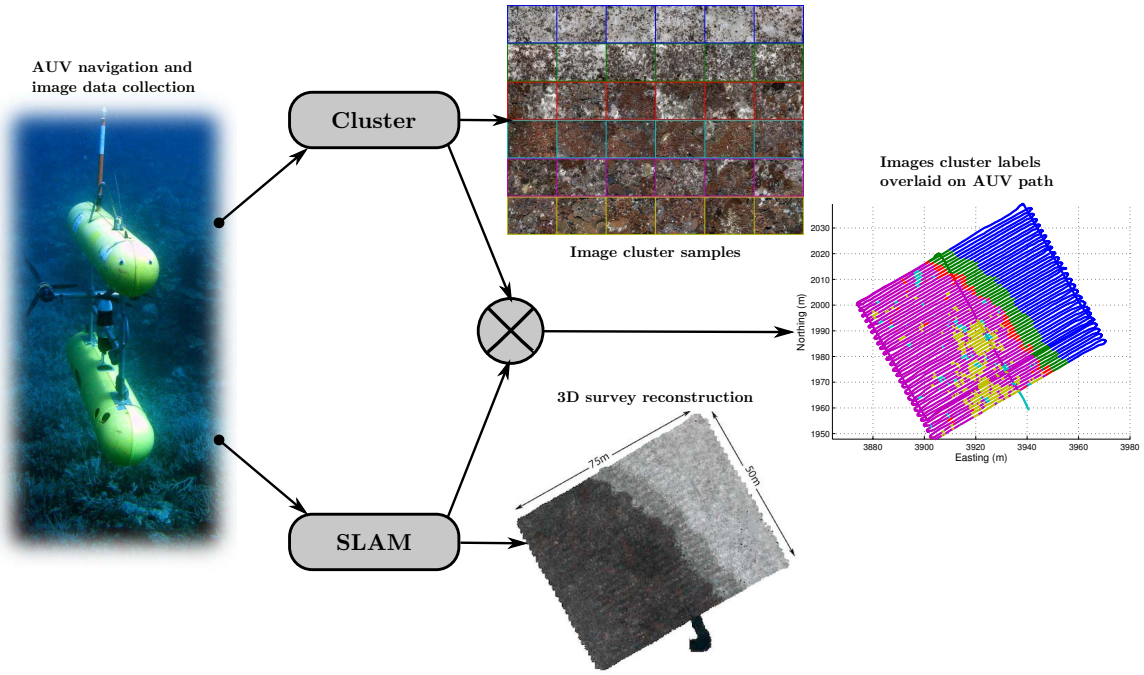


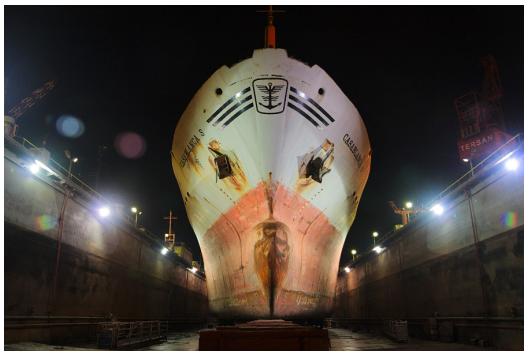
Figure 1.2 – An example automated data collection and summary pipeline. Providing coarse cluster labels and an estimate of where the images are in the environment is far more useful than simply providing imagery to marine ecologists. Simultaneous localisation and mapping (SLAM) is used to calculate an accurate vehicle trajectory for 3D environmental reconstruction. There are 10,000 images in this dive, which is from Scott Reef, Western Australia. This is a result from Steinberg et al. [103].

1.2 Problem Statement

An active field of research in the computer vision community is in supervised, or semi-supervised algorithms for high-level image "understanding". Image understanding refers to a more holistic attempt at automated interpretation and modelling of images as opposed to just object recognition or scene classification in isolation. This is an exiting and fast moving field. However, as mentioned previously there exist many real world applications where it is hard, or costly to obtain any amount of training data from a reliable source. To this end, this thesis is concerned with using, studying, and developing machine learning algorithms for completely *unsupervised* modelling of large collections of visual data.

An important aspect of supervised image understanding is utilising context to improve results. Context means certain objects, such as trees for example, usually occur

(a) Water dragon, near a creek.



(b) Cargo ship, in a ship yard.

Figure 1.3 – An example of objects in scene context. These images would not make as much sense if we were to swap the main objects in them. So the scene does give some information about the objects we would expect to find. But, can an unsupervised algorithm take advantage of these kinds of relationships, without any semantic understanding?

in specific places, such as outdoor scenes, and may have a particular location in an image. Figure 1.3 is also an example of this type of context. It can also mean certain types of scenes, such as forest, will mostly occur in particular collections of photos, such as hiking holidays. Achieving this type of contextual modelling in an unsupervised manner is a difficult problem. The primary aim of this work is to explore how successfully this can be done with standard machine learning frameworks. Another important aspect of this work is in uncovering how these contextual relationships work to improve results in this unsupervised setting.

A Bayesian approach to modelling is taken where possible. This is because it is relatively simple to construct models with certain hierarchical relationships using Bayesian techniques. Also, many of the more successful semi-supervised and supervised approaches to image understanding use hierarchical Bayesian models. Many of the algorithms presented are variations and/or novel applications of algorithms already in the computer vision and machine learning literature. However, particular emphasis is placed on exploring algorithm design decisions, for example, what effect certain prior distribution choices have on inference. And even more generally, how does choosing what to model in a collection of images affect performance in terms of computational complexity, and quality of labels/annotations compared to what a

human would generate?

The motivating applications usually involve inference over large datasets. For this reason, a lot of emphasis is placed on scalability of the algorithms, *even* if this means some quality of the end results have to be sacrificed. Consequently, a "meta" Bayesian model selection has been performed; preference is given to more simple algorithms that can achieve the stated goals satisfactorily. It is hoped that these analyses will also benefit the computer vision community at large.

1.3 Contributions

This thesis is primarily concerned with modelling large collections of visual data in a completely unsupervised manner. To this end, the contributions of this thesis arise from applying, adjusting and evaluating machine learning algorithms to specific instances of this overall problem. The principal contributions in this thesis are:

- Empirically demonstrating that popular unsupervised sparse coding and spatial pyramid pooling feature learning frameworks, such as that in [128], produce descriptors for images that are highly compressible with linear dimensionality reduction methods. This allows them to be used for large scale classification and clustering tasks. Similarly, the large over-complete dictionaries, or codebooks, learned by these techniques can generalise well to encoding novel/untrained datasets as long as the images used to train the dictionaries are diverse in appearance. This also facilitates large scale, and incremental learning applications.

- Developing a hierarchical Bayesian model for clustering multiple datasets that can take advantage of the natural partitioning of these datasets. It is shown that if multiple datasets are related in some way, such as photo albums of holidays, or separate AUV dives in one region, it is beneficial to share clusters between these datasets, while keeping the proportions of these clusters distinct to each dataset. These datasets can be seen as providing "context" for the observable

data within them. Or, more concretely, as providing different views of these clusters in feature space. This dramatically improves clustering performance and/or computational runtime. Furthermore, simple latent Dirichlet allocation (LDA)-like models [17] can be used to take advantage of this "context". Why this happens, and what model structures this works for are also explored.

- Extending these simple LDA-like models to exhibit *multi-level* clustering capabilities, that is, clustering images and image-parts simultaneously. Using these extended models, the effects of modelling context in an unsupervised fashion is explored further. For example, "objects" within images can be associated with specific image-type context, as well as dataset (album/survey) context. Also, these models can capture the co-occurrence of these "objects". These models are extended again so both observations of image parts, and whole images can be used to further improve clustering results. The advantages and disadvantages of each of these modelling choices are thoroughly explored and quantified. It is shown that rudimentary objects and scene types can be found efficiently using these *completely* unsupervised techniques.

1.4 Outline

Chapter 2 provides a brief overview of the precursors of the types of algorithms used in the thesis. Emphasis is placed on presenting clustering and topic-modelling algorithms, as these have largely influenced this work. A very brief primer on variational Bayes is presented, as well as an application of it to learning a basic Bayesian mixture model. This learning framework is used extensively in the thesis. Finally, an overview of the relevant computer vision literature is also presented, in increasing complexity from unsupervised and supervised scene classification, to multi-level scene understanding.

Chapter 3 is concerned with modifying popular unsupervised sparse single layer image feature learning frameworks, such as [128], for large scale clustering applications. A thorough empirical study is conducted to clarify; (a) which sparse coding

techniques provide a good trade-off between scalability and performance for whole image classification and clustering tasks. (b) How compressible image descriptors resulting from these frameworks are when using simple linear dimensionality reduction techniques. (c) How dependent image classification and clustering performance is on the choice of dictionary, or codebook, learning algorithm *and* training dataset. The contributions of this chapter are mainly intended for practitioners who want to use these techniques for large datasets, or incremental learning.

Chapter 4 starts to explore the advantages of using the "context" afforded by multiple related groups of data for clustering imagery. This chapter is motivated by the observation that AUV surveys, while spatially distinct, may share common habitat types, but in different proportions. For example, two surveys may both contain sand, but one also has kelp, while the other has no kelp, but does have visually similar sea-grass. These habitats may only get clustered into sand and "green stuff". However, sea grass and kelp probably don't often co-occur, and so modelling these datasets as distinct entities may help to disambiguate these visually similar, but contextually different, kelp and sea-grass clusters. Essentially this affords a clustering algorithm multiple views of the observations in which clusters of data may be better separated, or even absent, making cluster discovery easier. Modelling this structure with a simple Bayesian LDA-like mixture model is shown to improve the performance of clustering, and lessen computational runtime, compared to conventional clustering techniques. These algorithms are tested on two standard computer vision datasets, a large dataset from an AUV, and a novel photo albums dataset. It uses the image descriptors from Chapter 3.

Chapter 5 continues to explore the benefits of modelling various types of unsupervised context for clustering visual data. A Bayesian clustering model is developed that is inspired by some recently developed multi-level clustering models in the literature. As in Chapter 4, this clustering model also models the context that arises from multiple related datasets, but is extended to cluster both image parts (superpixels/segments) and images simultaneously. In this way, the type of image (image cluster) provides context for the image parts. This is another way of obtaining multi-

ple views of the observations (image parts) in feature space. The image-part clusters are analogous to "objects" (sky, trees, cows etc.), and the image clusters are formed from images that have similar proportions of these objects within them. The model from the previous chapter is also used to cluster image parts for comparison, but now uses images for context much like LDA is typically used, and has no notion of image clusters. Also a more conventional Bayesian clustering algorithm is used, which just clusters all image-parts together without context. The experiments in this chapter show enormous improvements in the algorithms that model some type of context compared to the conventional clustering algorithm. Also, the different methods of modelling context lead to some quite interesting results. For example, the more complex multi-level clustering algorithm is often the fastest of all algorithms computationally. Standard computer vision datasets, and an AUV dataset are used for these experiments.

Chapter 6 is concerned with correcting some of the limitations of the model presented in Chapter 5. For instance, it cannot easily be applied to large datasets, as it tends to over-cluster images. Also it does not seem to effectively take advantage of top-level albums or groups. A new model is proposed that uses both direct observations of image parts, like the previous model, *and* whole images. This is the only model in this thesis to make use of two observable variables. This model can effectively be used on larger datasets, and can also take advantage of top-level groups, while still providing a rich representation of the images. It is also shown to be as fast as the previous model, despite increased complexity. The model is tested on the same datasets from the previous chapter, as well as the larger photo albums dataset from Chapter 4.

Chapter 7 is a meta-conclusion of Chapters 3-6, and also provides a summary of potential future work and contributions following on from this thesis.

Chapter 2

Background

This chapter presents a brief background into the family of clustering, mixture and topic models in which the models derived later in this thesis belong. A brief primer on variational Bayes (VB) for learning these models is also given. Finally, a brief review of the relevant computer vision literature is presented, which provides context for the work in this thesis. Each section is self contained, so can be read on its own if the reader so desires.

2.1 An Overview of some Models for Unsupervised Learning

In this section a brief overview of some models for unsupervised learning, which underpin those derived in later chapters, is presented. Models for clustering and topic learning are the focus of this section, since they are fundamental to understanding the work in later chapters. While they are not presented here, it is also recommended the reader be familiar with some factor analysis and subspace learning models such as principal component analysis (PCA) [12, Ch. 12] and independent component analysis (ICA) [58]. These are commonly used for transforming data into a form more suitable for clustering.

2.1.1 Clustering

Clustering is one of the oldest data exploration methods. The objective is for an algorithm to discover sets of similar points, or observations, within a larger dataset. These sets are called *clusters*. Similarity is almost always characterised by some distance function between observations, such as Euclidean ℓ_2. Some of the more simple algorithms require the number of clusters to be specified in advance, while others can also infer this from the data, usually given other assumptions. K-means is one of the first and still most popular algorithms [77].

K-means

The objective of K-means clustering is to find K clusters of observations, within a dataset $\mathbf{X} = \{\mathbf{x}_n\}_{n=1}^N$, where $\mathbf{x}_n \in \mathbb{R}^D$. These clusters are characterised by their means, $\mathbf{M} = \{\boldsymbol{\mu}_k\}_{k=1}^K$ where $\boldsymbol{\mu}_k \in \mathbb{R}^D$. Each observation is assigned to a cluster mean using a label $z_n \in \{1, \ldots, K\}$, and $\mathbf{Z} = \{z_n\}_{n=1}^N$. The objective of K-means is to minimise the square loss, or reconstruction error,

$$\min_{\mathbf{M},\mathbf{Z}} \sum_{n=1}^{N} \sum_{k=1}^{K} \mathbf{1}[z_n = k] \|\mathbf{x}_n - \boldsymbol{\mu}_k\|_2^2. \tag{2.1}$$

Here $\mathbf{1}[\cdot]$ is an indicator function that evaluates to 1 when the condition in the brackets is true, and 0 otherwise. $\|\cdot\|_2$ is an ℓ_2 norm, or Euclidean distance. This is solved with two simple alternating steps. The first is the assignment step;

$$z_n = \arg\min_k \ \|\mathbf{x}_n - \boldsymbol{\mu}_k\|_2^2, \tag{2.2}$$

the next is the update step;

$$\boldsymbol{\mu}_k = \frac{\sum_n \mathbf{1}[z_n = k]\mathbf{x}_n}{\sum_n \mathbf{1}[z_n = k]}. \tag{2.3}$$

These two steps are iterated until the square loss in Equation 2.1 has converged.

Unfortunately this is not guaranteed to converge to a global minimum, and usually

many random initialisations (random choices of $\mathbf{x}_n$ for the initial $\boldsymbol{\mu}_k$) have to be attempted to find the best solution. This algorithm is very fast in practice though. Another disadvantage is that the number of clusters, K, has to be specified in advance. Perhaps more of a concern is that clusters are assumed to be essentially spherical because of the Euclidean distance used, which is quite often an over-simplification. It is also useful to have probabilistic assignments, $p(z_n = k|\mathbf{x}_n)$ rather than hard assignments. Gaussian mixture models solve these last two problems.

Gaussian Mixture Models

In a Gaussian mixture model (GMM), see Bishop [12], each observation is distributed according to a weighted sum of Gaussian distributions;

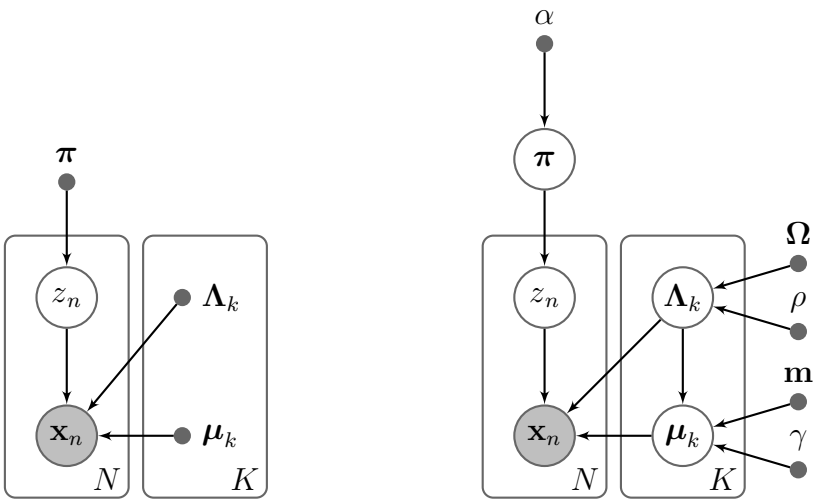
$$\mathbf{x}_n \sim \sum_{k=1}^{K} \pi_k \mathcal{N}\big(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k^{-1}\big)\,. \tag{2.4}$$

Here $\boldsymbol{\pi} = [\pi_1, \ldots, \pi_k, \ldots, \pi_K]^\top$ and $\pi_k \in [0, 1]$, with $\sum_k \pi_k = 1$. Also, Gaussian precision is used here instead of covariance ($\boldsymbol{\Lambda}_k^{-1} = \boldsymbol{\Sigma}_k$) for consistency later. What is still missing is a way to explicitly assign observations to mixtures or clusters. The same latent variable, z_n, used in K-means is introduced here as an auxiliary variable for this purpose, by inducing the following conditional relationship;

$$p(\mathbf{x}_n|z_n) = \prod_{k=1}^{K} \mathcal{N}\big(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k^{-1}\big)^{\mathbf{1}[z_n=k]}\,, \tag{2.5}$$

so given a cluster, $p(\mathbf{x}_n|z_k = k) = \mathcal{N}\big(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k^{-1}\big)$. Now it can be seen that each cluster is modelled as a single Gaussian, with a full covariance matrix. This auxiliary variable is itself distributed according to a Categorical distribution;

$$z_n \sim \mathrm{Cat}(\boldsymbol{\pi}) = \prod_{k=1}^{K} \pi_k^{\mathbf{1}[z_n=k]}. \tag{2.6}$$



(a) GMM **(b)** BGMM

Figure 2.1 – Graphical models of a GMM and Bayesian Gaussian mixture model (BGMM). Circles (nodes) are distributions, points are point estimates of parameters, and arrows (arcs) are conditional relationships. The shaded circle is observable, and plates denote replication over the respective index.

The joint, or "complete-data" likelihood is (omitting the conditional parameters),

$$p(\mathbf{X}, \mathbf{Z}) = \prod_{n=1}^{N} \mathrm{Cat}(z_n|\boldsymbol{\pi}) \prod_{k=1}^{K} \mathcal{N}\big(\mathbf{x}_n|\boldsymbol{\mu}_k, \mathbf{\Lambda}_k^{-1}\big)^{\mathbf{1}[z_n=k]} . \tag{2.7}$$

The graphical model of this joint is in Figure 2.1a. For a single observation, if the auxiliary variable is marginalised (summed) out of Equation 2.7, we are left with the marginal density in Equation 2.4.

Now we need an algorithm that can learn the labels, z_n, cluster parameters, $\boldsymbol{\mu}_k$ and $\mathbf{\Lambda}_k$, and mixture weights, $\boldsymbol{\pi}$. Such an algorithm can be derived by maximising (taking partial derivatives and setting to zero) the log-likelihood of the data, $\log p(\mathbf{X}) = \sum_n \log p(\mathbf{x}_n)$ from Equation 2.4, conditioned on the model parameters and latent variables. Firstly, maximising the log-likelihood with respect to z_n, yields;

$$p(z_n = k|\mathbf{x}_n) = \frac{1}{\mathcal{Z}_{z_n}} \pi_k \mathcal{N}\big(\mathbf{x}_n|\boldsymbol{\mu}_k, \mathbf{\Lambda}_k^{-1}\big) , \tag{2.8}$$

where $\mathcal{Z}_{z_n} = \sum_k \pi_k \mathcal{N}\big(\mathbf{x}_n|\boldsymbol{\mu}_k, \mathbf{\Lambda}_k^{-1}\big)$. This is known as the *expectation* step, since the

labels are assigned their expected value given the observations and cluster parameters. Next, the parameters can be found by maximising the log-likelihood with respect to each parameter;

$$\boldsymbol{\mu}_k = \frac{\sum_n p(z_n = k|\mathbf{x}_n)\,\mathbf{x}_n}{\sum_n p(z_n = k|\mathbf{x}_n)}, \tag{2.9}$$

$$\boldsymbol{\Lambda}_k^{-1} = \frac{1}{\sum_n p(z_n = k|\mathbf{x}_n)} \sum_{n=1}^{N} p(z_n = k|\mathbf{x}_n)\,(\mathbf{x}_n - \boldsymbol{\mu}_k)(\mathbf{x}_n - \boldsymbol{\mu}_k)^\top, \tag{2.10}$$

$$\pi_k = \sum_{n=1}^{N} \frac{p(z_n = k|\mathbf{x}_n)}{\sum_k p(z_n = k|\mathbf{x}_n)}. \tag{2.11}$$

This is called the *maximisation* step, because the value of the log-likelihood is maximised with respect to the parameters given the estimated latent variables. These two steps are iterated until the log-likelihood converges. This is known as the expectation maximisation (EM) algorithm, and as we can see, for all intents and purposes it is the same algorithm used to learn K-means. The exceptions being that a probabilistic assignment is learned, $p(z_n = k|\mathbf{x}_n)$, and *Mahalanobis*[1] distances are used (from the Gaussian clusters) as opposed to Euclidean distance. This allows clusters to have arbitrary ellipsoidal shapes. Furthermore, Gaussian clusters do not have to be used, for instance Multinomial clusters are another popular choice [17, 22, 79].

Unfortunately this algorithm has a few drawbacks. Like K-means, it is only guaranteed to converge to a local maximum of the likelihood function. Also, the Gaussian cluster updates require a full $D \times D$ covariance matrix inversion, which has a $\mathcal{O}\big(D^3\big)$ computational cost. This can be circumvented by using diagonal covariance Gaussian clusters, or other distributions such as Multinomial, that have only $\mathcal{O}\big(D\big)$ computational cost. Though some expressive power is lost since inter-dimensional correlation is not modelled.

Another drawback is that this algorithm still cannot choose K. One way to allow the EM algorithm to choose K is to include a penalty, or regulariser, for having too many parameters. In this way the maximum-likelihood fitting objective can be traded off

[1] $\text{dist}_{\text{Mahal.}} = (\mathbf{x}_n - \boldsymbol{\mu}_k)^\top \boldsymbol{\Lambda}_k (\mathbf{x}_n - \boldsymbol{\mu}_k)$

against a model complexity penalty. Some popular penalties are the Akaike information criterion (AIC) [3] and the Bayesian information criterion (BIC) [98]. These criterion tend to under-penalise model complexity [9], and are sometimes computationally costly to calculate. Another way to choose K is to use a fully Bayesian treatment, which in fact "averages" over all models to find the most simple model with the best fit for the data. In the case of mixture models, the learning algorithms can have very little additional computational cost compared to EM. For more details on maximum-likelihood GMMs, see [12, Ch. 9].

Bayesian Gaussian Mixture Models

In a Bayesian Gaussian mixture model (BGMM) [5, 12] conjugate prior distributions[2] are placed over all of the parameters in the maximum likelihood GMM model, for instance;

$$\boldsymbol{\pi} \sim \mathrm{Dir}(\alpha, \ldots, \alpha)\,, \tag{2.12}$$

$$\boldsymbol{\mu}_k \sim \mathcal{N}\big(\mathbf{m}, (\gamma\boldsymbol{\Lambda}_k)^{-1}\big)\,, \tag{2.13}$$

$$\boldsymbol{\Lambda}_k \sim \mathcal{W}(\boldsymbol{\Omega}, \rho)\,. \tag{2.14}$$

Here $\mathcal{W}(\cdot)$ is the Wishart distribution. The parameters over the parameters are called *hyper-parameters*. Also as shorthand, $\mathrm{Dir}(\alpha)$ may be used, which means the same hyper-parameter is used for each mixture weight. The joint distribution is now,

$$p(\mathbf{X}, \mathbf{Z}, \boldsymbol{\pi}, \mathbf{M}, \mathbf{L}) = \mathrm{Dir}(\boldsymbol{\pi}|\alpha) \prod_{k=1}^{K} \mathcal{N}\big(\boldsymbol{\mu}_k|\mathbf{m}, (\gamma\boldsymbol{\Lambda}_k)^{-1}\big)\, \mathcal{W}(\boldsymbol{\Lambda}_k|\boldsymbol{\Omega}, \rho) \times \prod_{n=1}^{N} \mathrm{Cat}(z_n|\boldsymbol{\pi}) \prod_{k=1}^{K} \mathcal{N}\big(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k^{-1}\big)^{\mathbb{1}[z_n=k]}\,, \tag{2.15}$$

[2] Conjugate priors, when multiplied or convolved with their corresponding likelihood distributions evaluate to a distribution with the same form as the prior. For instance, a Dirichlet prior, multiplied with a Multinomial likelihood evaluates to a posterior Dirichlet.

where $\mathbf{M} = \{\boldsymbol{\mu}_k\}_{k=1}^K$ and $\mathbf{L} = \{\boldsymbol{\Lambda}_k\}_{k=1}^K$. The graphical model of this joint is also in Figure 2.1b. Now, an important distinction in Bayesian learning is that the model parameters are *integrated out* for learning this model. Then the resulting log-*marginal* likelihood is maximised to derive the learning algorithm;

$$\log p(\mathbf{X}) = \log \int p(\mathbf{X}, \mathbf{Z}, \boldsymbol{\pi}, \mathbf{M}, \mathbf{L})\, d\mathbf{Z} d\boldsymbol{\pi} d\mathbf{M} d\mathbf{L}. \tag{2.16}$$

What is nice about this method is that a distribution over the models (as defined by the hyper-parameters) is learned from the data instead of one point estimate of the model. This allows the expected or average model to be used. The integral in Equation 2.16 is essentially an expectation over the latent variables and parameters, hence the term Bayesian "model averaging". Another way to think of this is that the marginal likelihood is the model evidence term (denominator) in Bayes' rule. Hence the model with the most evidence is chosen from a continuum of models, defined by the distributions over the model parameters. This model evidence term is, in essence, used to select the *best* number of clusters from the data. Practically, the number of clusters, K is chosen to be high, and some of the weights, $\boldsymbol{\pi}_k$, naturally fall off to 0 as learning progresses. Typically, the more data, the more evidence there is for an increased number of clusters.

Unfortunately, the integral in Equation 2.16 is, in general, intractable. Methods for approximating this integral must be resorted to in Bayesian inference. Some of the most popular of which are sampling methods, such as Markov chain Monte-Carlo (MCMC), and variational Bayes (VB) which is very similar to EM. VB is the chosen method used in [5] and in this thesis because it is deterministic, and almost as fast as EM for these simple mixture models. A primer on VB, and the general algorithm for mixture models, is given in Section 2.2.

The aforementioned BGMM is not the only Bayesian Gaussian mixture model. Another is the variational Dirichlet process (VDP) of Kurihara et al. [63]. Instead of using a Dirichlet distribution over the mixture weights as in Equation 2.12, it uses a Dirichlet process (DP) [36]. Describing a DP in depth is beyond the scope of this

thesis. Suffice to say it is a generalisation of the Dirichlet distribution to $K \to \infty$, i.e. it can represent an infinite number of clusters. Of course, using Bayesian learning only a finite K is found a-posteriori, depending on the evidence inherent in the data. Hence the DP is a Bayesian non-parametric process, since the number of parameters it has grows as the number of observations increase. The DP has a nice realisation under MCMC learning procedures, which distinguishes it from using a Dirichlet distribution. However, under VB learning there is less of an advantage, as a maximum truncation level of K is still typically chosen. There are larger model complexity penalties associated with a DP over a Dirichlet distribution, which has advantages for model selection in discrete mixture models – such as topic models [109]. However, unless there is only a very small amount of data to be clustered, the author has noticed almost no discernible difference between a regular BGMM and the VDP. Similar behaviour has also been noticed by Zobay [134]. They attribute this behaviour to the variational parameter updates being a strong function of the data-likelihood, and so the influence of the choice of prior is comparatively weak.

Similarity-Matrix Methods

There are numerous other clustering algorithms in the literature which are not based on K-means. Some popular methods use an $N \times N$ pair-wise similarity matrix between all points to cluster data. Typically similarity is defined to be some function of inverse distance (smaller distance, larger similarity). One method, which uses message passing between points in this matrix to find clusters is affinity propagation [44]. This is a simple and fast algorithm, which has also been modified to select the number of clusters. Unfortunately this model is similar to K-means in that it places a very restricted shape on the types of clusters it finds. Another very powerful method is spectral clustering by Ng et al. [81]. This creates a graph-Laplacian, or connectivity graph, out of the similarity matrix, then decomposes this graph spectrally using an Eigen solver. The "gaps" or differences between the largest Eigenvalues can be used to choose K. The corresponding K Eigenvectors are then clustered over their dimensionality N, using K-means. This algorithm has the advantage that it can

cluster arbitrary shapes. However, it lives or dies by how connected the original points are. In the case of highly overlapping/connected clusters this algorithm tends to perform very poorly compared to Gaussian mixture style models, which have strong assumptions about the data [119].

2.1.2 Topic Models

In this section a very brief overview of topic models is presented, with the intent of giving the reader a "flavour" of the field. Most of these models are not directly used in this thesis, however some of the models derived in later chapters have been influenced by these models.

The purpose of topic modelling is to generally perform inference on a large collection of textual documents, called a *corpus* [17, 109]. Inference may be to retrieve documents that are similar to various search terms (such as web search), or to discover collections of like documents from their distributions of words – essentially document clustering or classification.

Latent Semantic Analysis

One of the first topic models is latent semantic analysis (LSA), also known as latent semantic indexing (LSI) [87]. It describes a collection of documents as a matrix, $\mathbf{X} \in \mathbb{R}^{D \times J}$, whose columns are documents, and rows are transformed word frequency counts. There are many ways to create these transformed frequency counts, one of the more popular is to count the frequency of certain terms in each document, and then weight these term frequencies inversely by how often they occur in the corpus. Usually there are various logarithmic transformations and normalisations involved, but this basic idea is called term-frequency, inverse-document-frequency (*tf-idf*).

While this forms a compact representation of documents, it still may be very high-dimensional. LSA simply spectrally decomposes $\mathbf{X}\mathbf{X}^\top$ using singular value decomposition (SVD);

$$\mathbf{X}\mathbf{X}^\top = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top, \tag{2.17}$$

Where $\mathbf{U}$ are the Eigenvectors (cols) of $\mathbf{X}\mathbf{X}^\top$, $\mathbf{V}$ are the Eigenvectors (rows) of $\mathbf{X}^\top\mathbf{X}$, and $\mathbf{\Sigma}$ are the Eigenvalues. Now a reduced latent "semantic space" version of these documents can be found by projection,

$$\mathbf{Z} = \mathbf{\Sigma}_{1:K}^{-1}\mathbf{U}^\top{}_{1:K}\mathbf{X}, \tag{2.18}$$

where the $1:K$ subscript means only the top K Eigen-pairs are kept. Now these dimensionally reduced documents, $\mathbf{Z}$, can be compared, clustered etc. This is essentially PCA, except without mean-centring (subtraction of the mean from the data) of $\mathbf{X}$ to preserve sparsity.

While this model is computationally fast, it is not the best representation of words [56]. Essentially LSA assumes observations are Gaussian distributed (i.e. have positive and negative tf-idf counts), which is not observed in practice (a positive, discrete Poisson distribution is more realistic). This allows $\mathbf{\Sigma}$ and $\mathbf{U}$ to also have negative values, hence $\mathbf{Z}$ can be made up of negative linear combinations of tf-idf counts (i.e. $-0.3\times$car $+0.1\times$flower etc). This is somewhat nonsensical, and so $\mathbf{Z}$ has no direct semantic meaning, despite its name.

To overcome the limitations of LSA, Hofmann [56] formulated probabilistic latent semantic analysis (pLSA). It is quite different to LSA in that $\mathbf{X}$ is simply a count of the words (rows) in each document (cols), and does not inherently have a Gaussian assumption. It also introduces a latent variable that is essentially a distribution over "topics" given a document. These topics have a real semantic meaning, as opposed to the latent variables in LSA, since they are Categorical distributions over actual words. Each word in a document is drawn conditioned on a topic, and so there can be multiple topics in a document. Unfortunately this model is also quite limited in that the number of parameters that have to be learned (with EM) grows linearly in the number of documents. Also, it does not specify a proper generative model over documents, and so cannot generalise to new documents [17].

Latent Dirichlet Allocation

To rectify these problems with pLSA, Blei et al. [17] created latent Dirichlet allocation (LDA). It begins by defining a word, $x_{jn} \in \{1, \ldots, D\}$, as an index into a vocabulary of D word types. There are N_j words in a document, $\mathbf{X}_j = \{x_{jn}\}_{n=1}^{N_j}$, with J documents in a corpus, $\mathbf{X} = \{\mathbf{X}_j\}_{j=1}^{J}$. This is called a bag-of-words (BOW) model, because order of the words is assumed unimportant. This simplifying assumption is known as the exchangeability assumption.

LDA models words in a document as drawn from a *per-document* mixture of Categorical distributions,

$$x_{jn} \sim \sum_{k=1}^{K} \pi_{jk} \mathrm{Cat}(x_{jn}|\boldsymbol{\beta}_k)\,, \tag{2.19}$$

where $\boldsymbol{\beta}_k$ is also a vector of weights. These K categorical clusters are called "topics", and are *shared* between documents. What is nice about this model is that each document can now be described as a mixture of these K topics, $\boldsymbol{\pi}_j$. Typically $K \ll D$, and so this model is equivalent to discrete PCA [29], for dimensionality reduction. LDA can generalise to unseen documents, and the mixture weights can also be used in a similar fashion to the latent "semantic space" variable in LSA, while having a real semantic meaning.

This is a Bayesian model, and has a prior placed on each $\boldsymbol{\pi}_j \sim \mathrm{Dir}(\alpha)$, and sometimes a prior is also placed on $\boldsymbol{\beta}_k \sim \mathrm{Dir}(\phi)$, which is called smoothed LDA. The graphical model of smoothed LDA is presented in Figure 2.2.

LDA can use VB or sampling techniques, such as Gibbs sampling, for learning the model latent variables and hyper-parameters. Limitations have also been found with LDA. For example, it is not effective in choosing the number of topics (K), and the symmetric Dirichlet prior over topic weights, $\mathrm{Dir}(\boldsymbol{\pi}_j|\alpha)$, has been found to be too restrictive [109, 120].

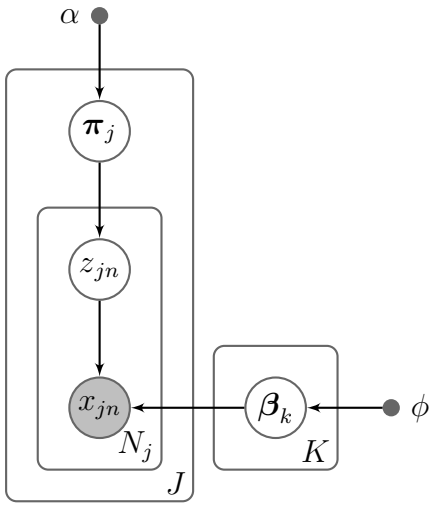


Figure 2.2 – Graphical model of smoothed LDA. This is very similar to a regular Bayesian mixture model, but replicated over J documents, with cluster, or topic, sharing between documents.

Hierarchical Dirichlet Process

To address some of the limitations of LDA, Teh et al. [109] introduced the hierarchical Dirichlet process (HDP). Explaining this model in depth is beyond the scope of this thesis, yet it is an interesting model and deserves a mention.

Essentially, a HDP replaces the Dirichlet prior on the document topic weights, $\boldsymbol{\pi}_j$, with a series of DPs with DP priors. A one-level HDP has a DP prior on the document weights, and then another DP prior over the DP on the document weights to enforce cluster or topic sharing between documents (without which, there would be no topic sharing, unlike a parametric prior, see [109] for details). A two-level HDP also places another DP prior on the model, which can then be used to model multiple corpora. These multiple corpora can also share topics, but the level of topic sharing can be controlled, i.e. some topics may be local to only some corpora.

While this is an incredibly flexible model, a DP is not a conjugate prior of another DP, and so closed-form updates do not exist. MCMC sampling can be used to learn the hyper-parameters of this model. A VB learning algorithm does also exist [110], but it is very complex because of this non-conjugate relationship.

Many other topic models exist which attempt to compensate for some of the shortcomings of LDA. For instance, the correlated topic model (CTM) by Blei and Lafferty [16], and Pachinko allocation model (PAM) by Li and McCallum [74] both model correlations between topics, and Steyvers et al. [104] models words as also being generated by authors. Despite this, LDA is still one of the most popular topic models.

2.2 Primer on Variational Bayes

Variational Bayes (VB) is the method of choice for learning all of the hierarchical Bayesian models in this thesis. The derivations of the learning steps for the algorithms are succinct in the following chapters since they are relatively straight forward once the VB framework is understood. This section is primarily for the benefit of those who are not familiar with the VB framework, and also to familiarise the reader with the notation used in the thesis. The VB framework will be presented from first principles for general latent variable models. Then a specific example of a derivation for a Bayesian exponential family mixture model is given, which is a precursor to the models used in this thesis.

2.2.1 Derivation of Variational Bayes

In this section a derivation of the general formulation of VB is given, largely following Beal [9], Bishop [12]. The objective is to tractably learn a model with latent variables ($\mathbf{Z}$ and $\boldsymbol{\Theta}$) which minimises the log-marginal likelihood,

$$\log p(\mathbf{X}) = \log \int p(\mathbf{X}|\mathbf{Z}, \boldsymbol{\Theta})\, p(\mathbf{Z}, \boldsymbol{\Theta})\, d\mathbf{Z} d\boldsymbol{\Theta}. \tag{2.20}$$

Here $\mathbf{X} = \{\mathbf{x}_n\}_{n=1}^N$ are observable variables, and $\mathbf{Z} = \{z_n\}_{n=1}^N$ are latent, or auxiliary, variables. These latent variables can assign the observable variables to mixture components in the case of mixture models, or can be latent states or homomorphic representations associated with each $\mathbf{x}_n$ in the case of dynamical and factor analysis

models respectively. The $\boldsymbol{\Theta} = \{\theta_k\}_{k=1}^K$ are latent model parameters, one per mixture, state etc., though these do not necessarily have to factor. An exact form for this model has not been specified, that is, the joint distribution in Equation 2.20 does not have any specified conditional independence between variables. This does not matter for the following derivation, however a simple form is explored in Section 2.2.2.

Free energy functional

In general, evaluating Equation 2.20 is intractable, particularly in the case of mixture models. This is because marginalising over all possible values for the latent variables/parameters is usually not feasible. Thus, an approximation to Equation 2.20 needs to be found. We start by approximating the posterior $p(\mathbf{Z}, \boldsymbol{\Theta}|\mathbf{X}) \approx q(\mathbf{Z}, \boldsymbol{\Theta})$, then re-cast Equation 2.20 as,

$$\log p(\mathbf{X}) = \log \int q(\mathbf{Z}, \boldsymbol{\Theta}) \frac{p(\mathbf{X}, \mathbf{Z}, \boldsymbol{\Theta})}{q(\mathbf{Z}, \boldsymbol{\Theta})} d\mathbf{Z} d\boldsymbol{\Theta}. \tag{2.21}$$

Using Jensen's Inequality, we can lower bound this log likelihood,

$$\log p(\mathbf{X}) \geq \int q(\mathbf{Z}, \boldsymbol{\Theta}) \log \frac{p(\mathbf{X}, \mathbf{Z}, \boldsymbol{\Theta})}{q(\mathbf{Z}, \boldsymbol{\Theta})} d\mathbf{Z} d\boldsymbol{\Theta} \tag{2.22}$$

with equality iff $p(\mathbf{Z}, \boldsymbol{\Theta}|\mathbf{X}) = q(\mathbf{Z}, \boldsymbol{\Theta})$. Now we apply a *mean field* approximation; that is, we assume all approximating distributions are independent, $q(\mathbf{Z}, \boldsymbol{\Theta}) \approx q(\mathbf{Z})\, q(\boldsymbol{\Theta})$, and only influence each other though some external "field"[3],

$$\begin{aligned}
\log p(\mathbf{X}) &\geq \int q(\boldsymbol{\Theta})\, q(\mathbf{Z}) \log \frac{p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\Theta})}{q(\mathbf{Z})} d\mathbf{Z} d\boldsymbol{\Theta} + \int q(\boldsymbol{\Theta}) \log \frac{p(\boldsymbol{\Theta})}{q(\boldsymbol{\Theta})} d\boldsymbol{\Theta}, \\
&= \mathbb{E}_{q_{\boldsymbol{\Theta}}, \mathbf{Z}} \left[\log \frac{p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\Theta})}{q(\mathbf{Z})} \right] + \mathbb{E}_{q_{\boldsymbol{\Theta}}} \left[\log \frac{p(\boldsymbol{\Theta})}{q(\boldsymbol{\Theta})} \right], \\
&= \mathcal{F}[q(\mathbf{Z}), q(\boldsymbol{\Theta})]. \qquad (2.23)
\end{aligned}$$

[3]These analogies come about from variational methods' original use in the study of physical systems, such as molecular fields [76].

Here, for instance, $\mathbb{E}_{q_{\mathbf{\Theta}}}[\cdot]$ means the expectation of the terms in the square brackets, with respect to the mean field distribution of $\mathbf{\Theta}$. $\mathcal{F}[\cdot]$ is known as the *free energy* functional or also as the evidence lower bound (ELBO). It lower-bounds the log marginal likelihood, $\log p(\mathbf{X})$, and allows for tractable optimisation of the latent variable distributions, $q(\mathbf{Z})$ and $q(\mathbf{\Theta})$, when we use *conjugate exponential* family models (more on this later). Optimising $\mathcal{F}[\cdot]$ is equivalent to minimizing the Kullback-Leibler (KL) between the approximate variational posterior and the true posterior. Essentially, this has recast Bayes' rule for determining the posterior into an optimisation problem!

Optimising for the Latent Variables

To find optimal values for the latent variables, $\mathbf{Z}$, *functional* derivatives can be taken of Equation 2.23 with respect to $q(\mathbf{Z})$,

$$\frac{\partial}{\partial q(\mathbf{Z})}\mathcal{F}[q(\mathbf{Z})\,, q(\mathbf{\Theta})] = 0. \tag{2.24}$$

Before we do this, we have to apply the constraint that $q(\mathbf{Z})$ is a valid probability density; sums to one and is non-negative. This is achieved using Lagrange multipliers to enforce $\int q(\mathbf{Z}) d\mathbf{Z} = 1$, and implicitly the $\log q(\mathbf{Z})$ term in $\mathcal{F}$ has to be greater than zero,

$$\frac{\partial}{\partial q(\mathbf{Z})}\left[\mathcal{F}[q(\mathbf{Z})\,, q(\mathbf{\Theta})] - \lambda\left(\int q(\mathbf{Z}) d\mathbf{Z} - 1\right)\right] = 0. \tag{2.25}$$

After some rearranging, the functional derivatives are taken of this constrained problem by using the Euler-Lagrange Equation (see Appendix B) to obtain,

$$\begin{aligned} 0 &= \int q(\mathbf{\Theta})\left[\log p(\mathbf{X}, \mathbf{Z}|\mathbf{\Theta}) - \log q(\mathbf{Z}) - 1 - \lambda\right] d\mathbf{\Theta}, \\ &= \int q(\mathbf{\Theta}) \log p(\mathbf{X}, \mathbf{Z}|\mathbf{\Theta})\, d\mathbf{\Theta} - \left[\log q(\mathbf{Z}) - 1 - \lambda\right] \int q(\mathbf{\Theta})\, d\mathbf{\Theta}. \end{aligned} \tag{2.26}$$

Noting that $\int q(\mathbf{\Theta}) d\mathbf{\Theta} = 1$, and defining $\log \mathcal{Z}_{\mathbf{Z}} = 1 + \lambda$ as a log normalisation constant,

$$\log q(\mathbf{Z}) = \int q(\mathbf{\Theta}) \log p(\mathbf{X}, \mathbf{Z}|\mathbf{\Theta})\, d\mathbf{\Theta} - \log \mathcal{Z}_{\mathbf{Z}},$$

$$= \mathbb{E}_{q_{\mathbf{\Theta}}}[\log p(\mathbf{X}, \mathbf{Z}|\mathbf{\Theta})] - \log \mathcal{Z}_{\mathbf{Z}}. \tag{2.27}$$

This yields the variational Bayes expectation (VBE) step of the latent variable $q(\mathbf{Z})$,

$$q(\mathbf{Z}) = \frac{1}{\mathcal{Z}_{\mathbf{Z}}} \exp\left\{\mathbb{E}_{q_{\mathbf{\Theta}}}[\log p(\mathbf{X}, \mathbf{Z}|\mathbf{\Theta})]\right\}. \tag{2.28}$$

Note how this step depends on expectations with respect to the variational model parameters, $\mathbf{\Theta}$.

Optimising for the Latent Parameters

In a similar fashion to the VBE step, we can find the model parameters by taking functional derivatives of Equation 2.23 with respect to $q(\mathbf{\Theta})$,

$$\frac{\partial}{\partial q(\mathbf{\Theta})} \mathcal{F}[q(\mathbf{Z}), q(\mathbf{\Theta})] = 0. \tag{2.29}$$

Again applying the constraint that $q(\mathbf{\Theta})$ is a valid density by using Lagrange multipliers to enforce $\int q(\mathbf{\Theta}) d\mathbf{\Theta} = 1$,

$$\frac{\partial}{\partial q(\mathbf{\Theta})} \left[\mathcal{F}[q(\mathbf{Z}), q(\mathbf{\Theta})] - \lambda \left(\int q(\mathbf{\Theta}) d\mathbf{\Theta} - 1\right)\right] = 0. \tag{2.30}$$

Taking functional derivatives and applying similar simplifications as in the VBE step,

$$\begin{aligned} 0 &= \int q(\mathbf{Z}) \left[\log p(\mathbf{X}, \mathbf{Z}|\mathbf{\Theta}) + \log p(\mathbf{\Theta}) - \log q(\mathbf{\Theta}) - 1 - \lambda\right] d\mathbf{Z}, \\ &= \int q(\mathbf{Z}) \log p(\mathbf{X}, \mathbf{Z}|\mathbf{\Theta})\, d\mathbf{Z} + \left[\log p(\mathbf{\Theta}) - \log q(\mathbf{\Theta}) - 1 - \lambda\right] \int q(\mathbf{Z})\, d\mathbf{Z}, \\ \log q(\mathbf{\Theta}) &= \int q(\mathbf{Z}) \log p(\mathbf{X}, \mathbf{Z}|\mathbf{\Theta})\, d\mathbf{Z} + \log p(\mathbf{\Theta}) - \log \mathcal{Z}_{\mathbf{\Theta}}. \end{aligned} \tag{2.31}$$

This yields the distribution over $\mathbf{\Theta}$ as given by the variational Bayes maximisation (VBM) step,

$$q(\mathbf{\Theta}) = \frac{1}{\mathcal{Z}_{\mathbf{\Theta}}} p(\mathbf{\Theta}) \exp\left\{\mathbb{E}_{q_{\mathbf{Z}}}[\log p(\mathbf{X}, \mathbf{Z}|\mathbf{\Theta})]\right\}. \tag{2.32}$$

Notice that Equation 2.32 includes a prior term over the parameters, $p(\boldsymbol{\Theta})$. This results in Bayesian updates over these parameters (prior×evidence), with the expectation fulfilling the role of the observation likelihood/evidence. Also, in contrast to the expectation step, expectation is with respect to the variational latent variables, $\mathbf{Z}$.

The Variational Bayes Algorithm

It can be seen that both Equation 2.28 and Equation 2.32 are dependent on one another, so in order to optimise $\mathcal{F}$ it is necessary to iterate between the VBE and VBM steps,

$$q(\mathbf{Z})^{(t+1)} = \frac{1}{\mathcal{Z}_{\mathbf{Z}}} \exp\left\{\mathbb{E}_{q_{\boldsymbol{\Theta}}^{(t)}}[\log p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\Theta})]\right\}, \tag{2.33}$$

$$q(\boldsymbol{\Theta})^{(t+1)} = \frac{1}{\mathcal{Z}_{\boldsymbol{\Theta}}} p(\boldsymbol{\Theta}) \exp\left\{\mathbb{E}_{q_{\mathbf{Z}}^{(t+1)}}[\log p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\Theta})]\right\}, \tag{2.34}$$

until the free energy converges to a local extremum, i.e. $(\mathcal{F}^{(t+1)} - \mathcal{F}^{(t)})/\mathcal{F}^{(t)} \to 0$.

This derivation used free energy to obtain the VB update steps, however it is useful to note the relationship between the log marginal likelihood, the KL divergence and free energy,

$$\log p(\mathbf{X}) = \mathrm{KL}[q(\mathbf{Z}, \boldsymbol{\Theta}) \| p(\mathbf{X}, \mathbf{Z}, \boldsymbol{\Theta})] + \mathcal{F}[q(\mathbf{Z}), q(\boldsymbol{\Theta})]. \tag{2.35}$$

It is also possible to derive the same VBE and VBM steps by taking functional derivatives with respect to the KL divergence [9, 12].

2.2.2 Application to a Conjugate Exponential Mixture Model

In this section the variational Bayes learning algorithm is derived for a simple and general Bayesian mixture model, with conjugate exponential family mixtures. The BGMM from the previous section belongs to this class of models. Also all of the novel models in this thesis use this model as their foundation, and so a basic understanding of this derivation is recommended.

Generative Model

In this model, the observable variables, $\mathbf{X} = \{\mathbf{x}_n\}_{n=1}^N$ where $\mathbf{x}_n \in \mathbb{R}^D$, are distributed according to a mixture of exponential family distributions;

$$\mathbf{x}_n \sim \sum_{k=1}^{K} \pi_k p(\mathbf{x}_n|\theta_k)\,. \tag{2.36}$$

Here $\boldsymbol{\pi} = [\pi_1, \ldots, \pi_k, \ldots, \pi_K]^\top$ and $\pi_k \in [0, 1]$, with $\sum_k \pi_k = 1$. Any distribution on the simplex can be used as a prior over $\boldsymbol{\pi}$, here we will use the simple symmetric Dirichlet;

$$\boldsymbol{\pi} \sim \mathrm{Dir}(\alpha, \ldots, \alpha)\,, \tag{2.37}$$

which can also be represented as $\mathrm{Dir}(\alpha)$. The observations, $\mathbf{x}_n$, can be drawn from any exponential family distribution given a mixture component k. Its parameters, θ_k, are drawn from a conjugate prior distribution with hyper-parameters η and $\boldsymbol{\nu}$,

$$p(\mathbf{x}_n|\theta_k) = f(\mathbf{x}_n)g(\theta_k)\exp\{\boldsymbol{\phi}(\theta_k)^\top \mathbf{u}(\mathbf{x}_n)\}, \tag{2.38}$$

$$p(\theta_k|\eta, \boldsymbol{\nu}) = h(\eta, \boldsymbol{\nu})g(\theta_k)^\eta \exp\{\boldsymbol{\phi}(\theta_k)^\top \boldsymbol{\nu}\}. \tag{2.39}$$

Here $g(\theta_k)$ and $h(\eta, \boldsymbol{\nu})$ are log-partition or normalisation functions, $\boldsymbol{\phi}(\theta_k)$ are natural parameters, $\mathbf{u}(\mathbf{x}_n)$ are sufficient statistics of the data, and $f(\mathbf{x}_n)$ is a function of $\mathbf{x}_n$. The conjugate exponential family includes many common distributions, such as Gaussian with Gaussian-Wishart priors, Multinomial or Categorical with Dirichlet priors, etc.

What is still missing is a way to explicitly assign observations to mixtures or clusters. To facilitate this, latent auxiliary variables, $\mathbf{Z} = \{z_n\}_{n=1}^N$, are introduced. These are discrete and take on the values $z_n \in \{1, \ldots, K\}$. The model is augmented with these auxiliary variables by introducing the following conditional relationship;

$$p(\mathbf{x}_n|z_n, \boldsymbol{\Theta}) = \prod_{k=1}^{K} p(\mathbf{x}_n|\theta_k)^{\mathbf{1}[z_n=k]}\,, \tag{2.40}$$

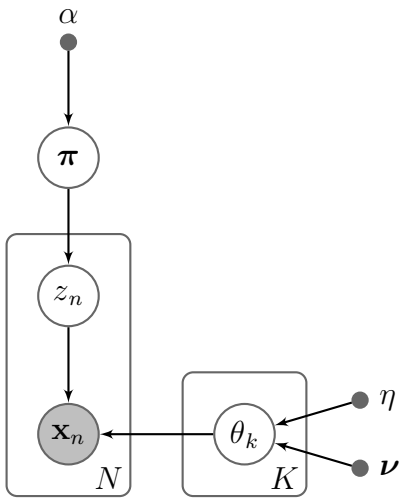


Figure 2.3 – Graphical model of a Bayesian mixture model. The shaded node is observable, and the points represent point estimates of the corresponding hyper-parameters. The plates denote replication over their respective index.

where $\mathbf{1}[\cdot]$ is an indicator function, and evaluates to 1 when the condition in the brackets is true, and 0 otherwise (this is essentially a Kronecker Delta). Also $\boldsymbol{\Theta} = \{\theta_k\}_{k=1}^K$. This auxiliary variable is distributed according to a Categorical distribution;

$$z_n \sim \mathrm{Cat}(\boldsymbol{\pi}) = \prod_{k=1}^{K} \pi_k^{\mathbf{1}[z_n=k]}. \tag{2.41}$$

Now an expression for the full joint distribution of the model can be found;

$$p(\mathbf{X}, \mathbf{Z}, \boldsymbol{\pi}, \boldsymbol{\Theta}) = \mathrm{Dir}(\boldsymbol{\pi}|\alpha) \prod_{k=1}^{K} p(\theta_k|\eta, \boldsymbol{\nu}) \prod_{n=1}^{N} \mathrm{Cat}(z_n|\boldsymbol{\pi})\, p(\mathbf{x}_n|z_n, \boldsymbol{\Theta})\,. \tag{2.42}$$

The graphical model of this factorised joint distribution is presented in Figure 2.3.

Variational Bayes Updates

The mean field approximation of the true posterior, $p(\mathbf{Z}, \boldsymbol{\pi}, \boldsymbol{\Theta}|\mathbf{X})$, is,

$$q(\boldsymbol{\pi}) \prod_{k=1}^{K} q(\theta_k) \prod_{n=1}^{N} q(z_n)\,. \tag{2.43}$$

From Equation 2.28 and following the style of Bishop [12], we can begin deriving the VBE step from the expectation,

$$\log q(\mathbf{Z}) = \mathbb{E}_{q_{\boldsymbol{\pi},\boldsymbol{\Theta}}}[\log p(\mathbf{X}, \mathbf{Z}, \boldsymbol{\pi}, \boldsymbol{\Theta})] - \log \mathcal{Z}_{\mathbf{Z}}. \tag{2.44}$$

Equation 2.28 does not use the full joint, unlike the above equation. This does not matter, because all of the parameter terms not involving $\mathbf{Z}$ are constants under this expectation, so can be lumped into $\mathcal{Z}_{\mathbf{Z}}$. Thus the two terms are equivalent. The full joint is used here, as in [12], because this exact step can be applied to all latent variables and parameters, as shall be seen later. So lumping all of the terms independent of $\mathbf{Z}$ into the normalisation constant, and factorising,

$$\log q(\mathbf{Z}) = \sum_{n=1}^{N} \mathbb{E}_{q_{\boldsymbol{\pi}}}[\log \mathrm{Cat}(z_n|\boldsymbol{\pi})] + \mathbb{E}_{q_{\boldsymbol{\Theta}}}[\log p(\mathbf{x}_n|z_n, \boldsymbol{\Theta})] - \log \mathcal{Z}_{\mathbf{Z}}. \tag{2.45}$$

From Equation 2.41 and Equation 2.40 both expectations factor over k,

$$\log q(z_n) = \sum_{k=1}^{K} \mathbf{1}[z_n = k] \cdot \mathbb{E}_{q_\pi}[\log \pi_k] + \mathbf{1}[z_n = k] \cdot \mathbb{E}_{q_\theta}[\log p(\mathbf{x}_n|\theta_k)] - \log \mathcal{Z}_{z_n}, \tag{2.46}$$

where $\mathbf{Z}$ has been implicitly factored over n. This can be used to evaluate the specific probability of an observation belonging to a cluster,

$$\begin{aligned} \log q(z_n = k) &= \mathbb{E}_{q_\pi}[\log \pi_k] + \mathbb{E}_{q_\theta}[\log p(\mathbf{x}_n|\theta_k)] - \log \mathcal{Z}_{z_n}, \\ q(z_n = k) &= \frac{1}{\mathcal{Z}_{z_n}} \exp\left\{\mathbb{E}_{q_\pi}[\log \pi_k] + \mathbb{E}_{q_\theta}[\log p(\mathbf{x}_n|\theta_k)]\right\}. \end{aligned} \tag{2.47}$$

This is basically the same as a maximum likelihood expectation step for a mixture model, except that the parameters $\boldsymbol{\pi}$ and θ_k are the (log) expected values, or model "averages", under their variational posterior distributions. These expectations are given in Appendix A (Dirichlet and Exponential Family sections), and the normali-

sation constant can be seen to be,

$$\mathcal{Z}_{z_n} = \sum_{k=1}^{K} \exp\left\{\mathbb{E}_{q_\pi}[\log \pi_k] + \mathbb{E}_{q_\theta}[\log p(\mathbf{x}_n|\theta_k)]\right\}. \tag{2.48}$$

Now for the VBM updates for $\boldsymbol{\pi}$. Again, we start with an expectation over the joint, just like in Equation 2.44,

$$\log q(\boldsymbol{\pi}) = \mathbb{E}_{q_{\mathbf{Z},\boldsymbol{\Theta}}}[\log p(\mathbf{X}, \mathbf{Z}, \boldsymbol{\pi}, \boldsymbol{\Theta})] - \log \mathcal{Z}_{\boldsymbol{\pi}}. \tag{2.49}$$

Note how only the expectations and normalisation constants have changed, and the expectation are with respect to all latent variables but $\boldsymbol{\pi}$. Again expanding, and lumping all terms independent of $\boldsymbol{\pi}$ into the normalisation constant,

$$\begin{aligned}
\log q(\boldsymbol{\pi}) &= \mathbb{E}_{q_{\mathbf{Z},\boldsymbol{\Theta}}}\left[\log \mathrm{Dir}(\boldsymbol{\pi}|\alpha) + \sum_{n=1}^{N} \log \mathrm{Cat}(z_n|\boldsymbol{\pi})\right] - \log \mathcal{Z}_{\boldsymbol{\pi}}, \\
&= \log \mathrm{Dir}(\boldsymbol{\pi}|\alpha) + \sum_{n=1}^{N} \mathbb{E}_{q_z}[\log \mathrm{Cat}(z_n|\boldsymbol{\pi})] - \log \mathcal{Z}_{\boldsymbol{\pi}}, \\
q(\boldsymbol{\pi}) &= \frac{1}{\mathcal{Z}_{\boldsymbol{\pi}}} \mathrm{Dir}(\boldsymbol{\pi}|\alpha) \prod_{n=1}^{N} \exp\left\{\mathbb{E}_{q_z}[\log \mathrm{Cat}(z_n|\boldsymbol{\pi})]\right\}.
\end{aligned} \tag{2.50}$$

We can see this has the same form as the VBM step in Equation 2.32. It also looks suspiciously like a conjugate arrangement, with only the expectation causing some complications, so expanding this using Equation 2.41,

$$\begin{aligned}
q(\boldsymbol{\pi}) &= \frac{1}{\mathcal{Z}_{\boldsymbol{\pi}}} \mathrm{Dir}(\boldsymbol{\pi}|\alpha) \prod_{n=1}^{N} \exp\left\{\mathbb{E}_{q_z}\left[\sum_{k=1}^{K} \mathbf{1}[z_n = k] \log \pi_k\right]\right\}, \\
&= \frac{1}{\mathcal{Z}_{\boldsymbol{\pi}}} \mathrm{Dir}(\boldsymbol{\pi}|\alpha) \prod_{n=1}^{N} \exp\left\{\sum_{k=1}^{K} q(z_n = k) \log \pi_k\right\}, \\
&= \frac{1}{\mathcal{Z}_{\boldsymbol{\pi}}} \mathrm{Dir}(\boldsymbol{\pi}|\alpha) \prod_{n=1}^{N} \prod_{k=1}^{K} \pi_k^{q(z_n=k)},
\end{aligned} \tag{2.51}$$

where $\mathbb{E}_{q_z}[\mathbf{1}[z_n = k]] = q(z_n = k)$. If we expand all terms out, and solve, it is straight

forward to find that $q(\boldsymbol{\pi})$ is a "posterior" Dirichlet,

$$q(\boldsymbol{\pi}) = \mathrm{Dir}(\boldsymbol{\pi}|\tilde{\alpha}_1, \ldots, \tilde{\alpha}_k, \ldots, \tilde{\alpha}_K)\,, \tag{2.52}$$

where,

$$\tilde{\alpha}_k = \alpha + \sum_{n=1}^{N} q(z_n = k)\,. \tag{2.53}$$

These variational posterior hyper parameters are really just the priors with pseudo-observation counts. The normalisation constant, $\mathcal{Z}_{\boldsymbol{\pi}}$ can be simply found by recognising that $q(\boldsymbol{\pi})$ is a Dirichlet distribution.

The same procedure can be followed for the VBM updates to $\boldsymbol{\Theta}$. Starting with an expectation over the joint.

$$\log q(\boldsymbol{\Theta}) = \mathbb{E}_{q_{\mathbf{Z},\boldsymbol{\pi}}}[\log p(\mathbf{X}, \mathbf{Z}, \boldsymbol{\pi}, \boldsymbol{\Theta})] - \log \mathcal{Z}_{\boldsymbol{\Theta}}. \tag{2.54}$$

The result is very similar to $\boldsymbol{\pi}$, where after a bit of factoring over k,

$$q(\theta_k) = p(\theta_k|\tilde{\eta}_k, \tilde{\boldsymbol{\nu}}_k)\,, \tag{2.55}$$

and,

$$\tilde{\eta}_k = \eta + \sum_{n=1}^{N} q(z_n = k)\,, \tag{2.56}$$

$$\tilde{\boldsymbol{\nu}}_k = \boldsymbol{\nu} + \sum_{n=1}^{N} q(z_n = k)\,\mathbf{u}(\mathbf{x}_n)\,. \tag{2.57}$$

Again we see pseudo observation counts for $\tilde{\eta}_k$, and also weighted sufficient statistic contributions for $\tilde{\boldsymbol{\nu}}_k$.

The Variational Lower Bound

While only the VBE and VBM steps are required to learn a model, it is also useful to calculate the free energy, $\mathcal{F}$, explicitly for monitoring convergence. It can also

be used to aid in model selection, e.g. choosing the optimal number of mixtures, K, when using heuristics. *Negative* free energy is usually used in this thesis (which is usually just referred to as $\mathcal{F}$), following [9] – purely for aesthetic reasons. Using Equation 2.23,

$$
\begin{aligned}
-\mathcal{F}[q(\mathbf{Z}), q(\boldsymbol{\pi}), q(\boldsymbol{\Theta})] = \mathbb{E}_{q_{\boldsymbol{\pi}}}\left[\log \frac{q(\boldsymbol{\pi})}{\mathrm{Dir}(\boldsymbol{\pi}|\alpha)}\right] + \sum_{k=1}^{K} \mathbb{E}_{q_\theta}\left[\log \frac{q(\theta_k)}{p(\theta_k|\eta, \boldsymbol{\nu})}\right] \\
+ \sum_{n=1}^{N} \mathbb{E}_{q_{\mathbf{Z},\boldsymbol{\pi},\boldsymbol{\Theta}}}\left[\log \frac{q(z_n)}{\mathrm{Cat}(z_n|\boldsymbol{\pi})\, p(\mathbf{x}_n|z_n, \boldsymbol{\Theta})}\right] \quad (2.58)
\end{aligned}
$$

In this model, the last term involving z_n actually simplifies to $-\sum_n \log \mathcal{Z}_{z_n}$. This is neat, since it is natural to see the trade off between fitting data in the $\mathcal{Z}_{z_n}$ term, and a complexity penalty arising from the difference between the entropy of the variational posterior and cross-entropy of encoding the posterior parameters under the prior parameter distributions. For example,

$$
\begin{aligned}
\mathbb{E}_{q_{\boldsymbol{\pi}}}\left[\log \frac{q(\boldsymbol{\pi})}{\mathrm{Dir}(\boldsymbol{\pi}|\alpha)}\right] &= \mathbb{E}_{q_{\boldsymbol{\pi}}}[\log q(\boldsymbol{\pi})] - \mathbb{E}_{q_{\boldsymbol{\pi}}}[\log \mathrm{Dir}(\boldsymbol{\pi}|\alpha)] \\
&= -\mathrm{H}(\boldsymbol{\pi}) + \mathrm{H}_{q\|p}(\boldsymbol{\pi}) \quad (2.59)
\end{aligned}
$$

All of the expectations here are given in Appendix A.

The Algorithm

In this section the explicit VB algorithmic steps are presented to aid understanding of how this learning algorithm is implemented. See Algorithm 2.1 for an example implementation of the Bayesian exponential mixture model VB algorithm.

In Algorithm 2.1 an initial truncation level, K^*, for the number of clusters is used. As learning proceeds a number of these clusters become empty (they have less than one observation) and so the actual number of clusters, $K \leq K^*$, is found automatically. Random initialisation for the labels is not always essential, but it does make for a more simple implementation. In later chapters deterministic cluster splitting and

Algorithm 2.1: The Bayesian exponential mixture model VB algorithm

Data: Observations $\mathbf{X}$, and an (over) estimate K^*
Result: Probabilistic assignments $q(\mathbf{Z})$, and posterior hyper-parameters $\{\tilde{\alpha}_k, \tilde{\eta}_k, \tilde{\boldsymbol{\nu}}_k\}_{k=1}^{K}$

$\{\alpha, \eta, \boldsymbol{\nu}\} \leftarrow$ `CreatePriors()`; `// e.g.` $\alpha = 1$
$q(\mathbf{Z}) \leftarrow$ `RandomLabels(`K^*`)`;
$\mathcal{F} \leftarrow$ some large number;

repeat
  $\mathcal{F}_{old} \leftarrow \mathcal{F}$;
  $\{\tilde{\alpha}_k, \tilde{\eta}_k, \tilde{\boldsymbol{\nu}}_k\}_{k=1}^{K^*} \leftarrow$ `VBMaximisation(`$\mathbf{X}, q(\mathbf{Z})$`);` `// Eqns. 2.53, 2.56 and 2.57`
  $q(\mathbf{Z}) \leftarrow$ `VBExpectation(`$\mathbf{X}$, $\{\tilde{\alpha}_k, \tilde{\eta}_k, \tilde{\boldsymbol{\nu}}_k\}_{k=1}^{K^*}$`);` `// Equation 2.47`
  $\mathcal{F} \leftarrow$ `VBLowerBound(`$q(\mathbf{Z})$, $\{\alpha, \eta, \boldsymbol{\nu}\}$, $\{\tilde{\alpha}_k, \tilde{\eta}_k, \tilde{\boldsymbol{\nu}}_k\}_{k=1}^{K^*}$`);` `// Equation 2.59`
until $(\mathcal{F}_{old} - \mathcal{F})/\mathcal{F}_{old} < C_{threshold}$;

$q(\mathbf{Z})$, $\{\tilde{\alpha}_k, \tilde{\eta}_k, \tilde{\boldsymbol{\nu}}_k\}_{k=1}^{K}$, $\leftarrow$ `RemoveEmptyClusters(`$q(\mathbf{Z})$, $\{\tilde{\alpha}_k, \tilde{\eta}_k, \tilde{\boldsymbol{\nu}}_k\}_{k=1}^{K^*}$`);`

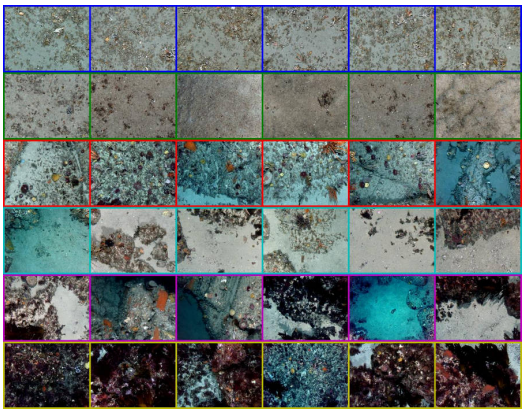
search heuristics are used instead when appropriate.

This concludes the derivation of a general exponential family Bayesian mixture, and the reader is referred to Bishop [12, Ch. 10] and Beal [9, Ch. 2] for more information on these models.

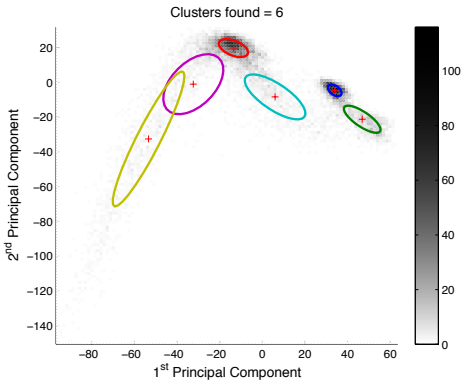
2.3 Modelling Visual Data – Literature Review

This section presents a brief overview of some of the computer vision literature that is related to the work in this thesis. This is mostly to set the stage for this work, and is not intended to be exhaustive.

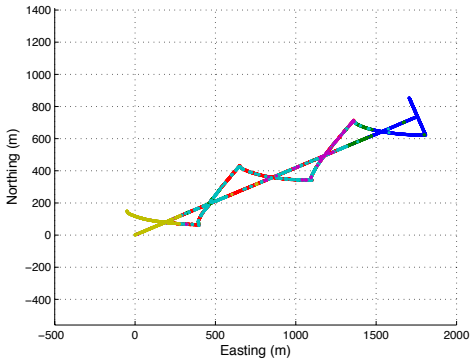
This overview starts from unsupervised and supervised scene recognition, and then proceeds to literature dealing with object recognition and detection within images. Finally a brief overview is given of the research into more holistic scene understanding, from low level objects to high level scene categories, using associated tags and captions when available.



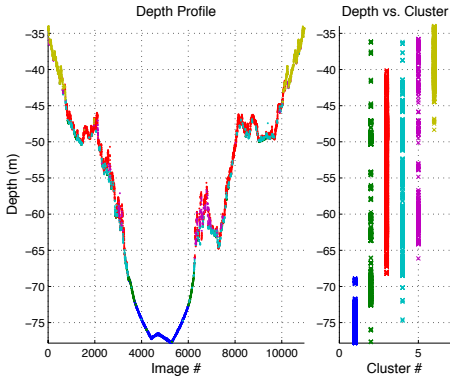
(a) Random image samples.



(b) Gaussian clusters.



(c) Top-down view of survey.



(d) Depth vs. cluster.

Figure 2.4 – Sample results of clustering imagery from an autonomous underwater vehicle (AUV) from Steinberg et al. [103]. The VDP was used to cluster images based on colour, texture and 3D morphology features. Random images from each of the clusters shown in (a). Images and Gaussian clusters are shown in the first two principal dimensions of feature space in (b). Images colour by cluster memberships are overlaid on the AUV transect in (c), and plotted verse depth in (d). Note the cluster correlation with depth, despite this not being explicitly modelled. The kelp cluster appears in the more shallow, photic zone.

2.3.1 Unsupervised Scene Recognition

Unsupervised scene recognition is one of the more fundamental tasks in computer vision. Given a collection of images, can an algorithm find clusters of like images? The number of clusters may be chosen a-priori, like in K-means or maximum-likelihood mixtures [79, 117]. Alternatively fully Bayesian non-parametric approaches have been used to solve this problem of choosing the number of clusters [50, 103].

Of primary importance in using these data-driven methods is choosing a highly discriminating way to describe images. These are referred to as image features or descriptors. Masada et al. [79] use quantised colour histograms, and simple spatial information from wavelet transforms as image descriptors. Multinomial and Bayesian Multinomial-Dirichlet conjugate pair mixtures were used to model these image descriptors. This work is effective in separating scenes based on these simple cues, but it unable to distinguish between images of red flowers with grass background, and London buses in front of trees for instance.

Steinberg et al. [103] also use simple colour statistics, as well as texture histograms from local binary patterns [83], and 3D scene morphology features from Friedman et al. [45]. These image descriptors are modelled as a Bayesian non-parametric Gaussian mixture – the VDP of Kurihara et al. [63]. The descriptors appear to work well on the underwater imagery they were designed for. Some examples are shown in Figure 2.4. However, they are less useful on more highly structured imagery, such as scenery.

Pizarro et al. [90] also have a similar objective to [103], in that clusters of similar underwater imagery are to be found. However their approach is quite different. A bag-of-words (BOW) image description is used, where scale-invariant feature transform (SIFT) features [75] are extracted from all images, either at interest-points or in a dense grid, and are quantised by K-means. The quantised SIFT descriptors are the "words". These may then be pooled (summed) per image (bag), providing the image descriptor, or can be combined with topic models to find a more compact image descriptor. In [90] the pooled BOW descriptors are clustered directly using symmetric KL divergence in an agglomerative, hierarchical clustering scheme. Also, the BOW descriptors are dimensionally reduced using LDA before clustering, which speeds inference. Both tactics produce a hierarchy of image clusters, which at many levels look very consistent. Girdhar et al. [48] also use topic models for underwater unsupervised scene recognition. Their goal in this instance is to find novel images, defined by images that are distant in the latent topic space. They use the novelty of the images captured by a small AUV to guide its behaviour – it will linger over novel

scenes, for instance.

Gomes et al. [50] also use an incremental variant of the VDP [63] used in [103]. They use the spatial pyramid match kernel from [65] with kernel-principal component analysis (k-PCA) for dimensionality reduction to describe images. This appears to work well on a 4-class subset of Caltech-256 [52], though with so few classes it is hard quantify how well this approach would work on a more diverse dataset.

Tuytelaars et al. [117] provide a fairly exhaustive comparison between clustering methods; K-means, spectral methods, and topic models such as LDA and non-negative matrix factorisation (NMF) for "object" discovery. Though they use subsets of Caltech-256 [52], which only has one object per image, and so is similar to scene clustering. A BOW image representation is also used. They see no benefit in topic models over clustering for single object discovery tasks, and find that simple K-means is very competitive with the other methods.

2.3.2 Supervised Scene Recognition

Supervised image recognition, or classification, is another fundamental task in computer vision. The aim is now to generalise human provided labels to unseen/unlabelled images accurately. Early work by Torralba and Oliva [114] used second order statistics from the spectra of natural images as image descriptors. Then a supervised Gaussian mixture model (similar to a naïve Bayes classifier) was used to classify these images. It was reasonably effective at discriminating between natural and man-made scenes, scenes with and without animals, etc.

Some relatively successful attempts at more complex scene classification also use a BOW image representation with topic models such as pLSA [21] or LDA [41] to create a low-dimensional descriptor for each image. These topic models themselves can be modified to classify images, i.e., one topic model per class where a test image is assigned to the model with maximum likelihood. Or they could simply provide descriptors for other classifiers, such as support vector machines (SVMs). All of these models learn the image descriptors in an unsupervised fashion.

More recently, many state of the art methods generalise the BOW methods to account for the spatial layout of the image features, such as the spatial pyramid match kernel of Lazebnik et al. [65]. This method uses vector quantisation (VQ) with SIFT descriptors, as well as a histogram intersection function within subdivided regions of an image. These histogram intersection functions are combined into the spatial pyramid matching kernel, preserving their spatial layout. Classification was then performed using a SVM with this kernel, and provided performance significantly greater than the state-of-the-art methods at the time. This work was generalised further by Yang et al. [128] to use sparse coding with SIFT descriptors as opposed to VQ (more on this in Chapter 3). It also used max-pooling to combine these codes into a "spatial pyramid". This method was called sparse code spatial pyramid matching (ScSPM), and allowed simple and fast linear kernel SVMs to obtain better classification performance in less time than [65].

Current state of the art methods are often based on [128], such as [25], or use similar concepts, but in multi-layered or deeper networks [11, 20, 67]. In these multi-layered networks, many coding and pooling stages may be chained together. Each layer may learn more robust and invariant image representations, before the final image descriptors are classified.

2.3.3 Object Recognition and Discovery

Object recognition and discovery is a fairly broad area, encompassing supervised and unsupervised methods for segmenting, localising, and retrieving single and multiple objects within images. Typically objects refers to concrete things like tree, faces, cars, and more nebulous things like sky, cities, forests, water etc.

Some of the earlier and more simple methods (already mentioned in passing) [114, 117], focused on classifying, or clustering, single objects per image – and so in some ways are similar to scene recognition. These methods are usually not concerned with the object's location within an image. More advanced methods attempt to classify, and detect an object's location within an image using various descriptors

such as SIFT with nearest-neighbour classifiers [75], and mixtures of constellation models [42]. These methods also describe an object as a collection of more simple parts, and the method described in [75] can detect multiple objects within a scene.

Unsupervised object segmentation has not been given much attention in the literature, with single-object clustering methods [50, 101, 117] being more prevalent. An exception being Russell et al. [95]. The work of Sivic et al. [101] is a good example of single-object unsupervised object recognition and segmentation using BOW features and hierarchical-LDA [13]. It has also been noted by Tuytelaars et al. [117] that unsupervised recognition of multiple objects per scene is typically very difficult, and remains a largely unsolved problem. Russell et al. [95] use multiple segmentation results per image combined with topic models, such as pLSA and LDA, in order to discover the "best" object segmentations in an unsupervised manner. The resulting objects are very visually consistent. Some further attention has been given to this problem since, but it is usually in the guise of semi-supervised learning (and scene understanding), where some labelled data, or related textual information, is available [39].

It has also been found that context plays a huge role in improving the identification of objects. For instance, an example given by Torralba et al. [115] is that you would typically only find a coffee machine in a kitchen. Consequently, scene recognition may be used in conjunction with object detection, as in [115]. There they 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. Similarly it has been noted that objects commonly co-occur, and so detection of one object (street) may be used to aid detection of another object (building). This has been demonstrated by Choi et al. [31]. They use tree-like models to infer the contextual and spatial relationships of and between labelled objects to aid inference in unlabelled test sets.

Naturally modelling these more complex interactions between parts of images leads to models that attempt to more fully "understand" images.

2.3.4 Scene Understanding

With the realisation that context plays a large part in object and scene recognition, and with the advent of textual data often accompanying images on the Internet in the form of tags, and captions, much of the recent literature has focused on holistic image "understanding".

Some of the first attempts at this used modified topic models [30, 39]. Both of these models cluster super-pixels from over-segmented images, with the option of classifying the image if given labelled examples. The features used to represent each image are the proportions of super-pixel clusters (objects) within the images. The super-pixels are usually described by a combination of BOW features within the super-pixels, and quantised super-pixel attributes like colour and texture. A notion of the relative spatial layout or contiguity of the super-pixels is also modelled by Du et al. [39]. Their model can also work in a fully unsupervised setting, but they do not show complete results of the super-pixel and image clusters in this situation, preferring to show only the annotated examples of the super-pixel clusters.

Fei-Fei and Li [40] 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.

Li et al. [72] 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 tags 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. Other works, such as [14, 73, 102, 113] also try to learn commonalities between images and their associated textual corpora, in order to improve inference

for both tasks.

More recently, Li et al. [69] combined sparse-code dictionary learning and encoding, 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.

Niu et al. [82] do not model the text associated with images, apart from scene or object labels. They present a model that is in principal similar to [30], but it also learns the absolute position of objects within a scene type. It learns that a sky object is at the top of an image, buildings at the sides in a street scene 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.

This is a very active area of research, and only a small part of the literature has been mentioned here. But, as can be seen from this overview, much of the literature is concerned with supervised or semi-supervised image understanding. This overview also exemplifies that there is a lot to be gained by using contextual cues for image understanding. The work presented in the upcoming chapters leverages this reviewed literature, and makes inroads into a more fully-fledged unsupervised image understanding framework in the absence of any annotations or related textual information.

Chapter 3

Adapting Feature Learning for Large Scale Image Clustering

The performance of any supervised classification and unsupervised clustering algorithm is bounded by the quality of the method used to describe the observations. The aim of this chapter is to compare some of the more successful feature learning techniques present in the supervised image classification literature for the purposes of unsupervised image clustering. Simple and fast one-layer feature learning techniques are preferred since they are easily implemented and scalable to large datasets.

A contribution of this chapter is empirically demonstrating that existing single-layer, sparse coding feature learning frameworks learn image descriptors that are not only linearly separable, but are also very compressible with linear dimensionality reduction methods. When compressed, these descriptors are shown to perform almost as well as the uncompressed descriptors in classification tasks. These compressed descriptors and are also shown to be suitable for clustering algorithms. Another contribution is in empirically demonstrating that over-complete dictionaries with a diverse set of elements can generalise well to new datasets, that have not been trained on, for both classification and clustering tasks when used in these feature learning pipelines. Ultimately it is shown that it is relatively straightforward to adapt these feature learning frameworks to scale to large datasets with little engineering effort.

3.1 Introduction

Recently there has been much interest in biologically inspired sparse feature learning systems used for applications such as classification of imagery [20, 33, 34, 65, 121, 128], reconstruction of signals in the presence of noise or missing data [2, 133], and compressive sensing [38, 133]. These techniques have all achieved state of the art results in their respective applications. They are particularly attractive in scene classification applications since the sparse features that are learned, though high dimensional, are easily separable with fast and scalable linear kernel support vector machines (SVMs) [121, 128].

An important part of sparse feature learning is learning a usually over-complete[1] set of vectors that spans the space of all observed signals. These vectors are often referred to as filters, elements or bases, though the latter is a misnomer since they are typically not linearly independent. The set of vectors is called a dictionary, codebook or frame, and is used to encode observations. It is typical in the literature to learn this dictionary from the same dataset that is classified, de-noised, in-painted etc. In this chapter one aim is to quantify how generalisable the popular single layer sparse code spatial pyramid matching (ScSPM) framework, introduced by Yang et al. [128], is to representing *unseen* data, without having to relearn the dictionary. Another aim is to quantify whether this framework is suited to providing features for clustering applications – thereby allowing for completely unsupervised image "understanding" frameworks.

Coates and Ng [33], demonstrate that as long as a dictionary is sufficiently over-complete and its elements sufficiently diverse[2], most of the classification performance of ScSPM sparse-coding framework is actually due to the encoding method. Boureau et al. [25, 26] theoretically and empirically examine the effects of the spatial pyramid pooling, and suggest that this may account for more performance increase than even the choice of encoding method. The experiments presented in this chapter extend upon these results by showing that as long as the dataset used to train the dictio-

[1] That is, there are more vectors than dimensions, resulting in an over-determined system.
[2] How to actually measure this is a field of research unto itself [38].

nary is diverse, i.e. it effectively "tiles" the space characterised by its dimension, the resulting dictionary could be applied to novel datasets without a considerable reduction in performance to classification and clustering. This opens up the possibility of learning one dictionary on a diverse training set, and then applying it to unseen data, potentially for incremental applications.

It is a well known fact that many clustering algorithms based on distance or similarity measures between data points are susceptible to the curse of dimensionality. Furthermore, clustering methods based on Mahalanobis distance metrics (some forms of K-means, Gaussian mixture models) may require inversion of covariance matrices. This inversion has a $\mathcal{O}(D^3)$ computational cost, where D is dimension. For these reasons, it is common to use dimensionality reduction techniques prior to clustering. Since sparse codes, when combined with pooling techniques such as in [25, 33, 128], are separable with linear kernel SVMs, in this thesis it is conjectured and then empirically shown that they are also compressible with simple linear techniques such as principal component analysis (PCA). This then allows for the use of these sparse image representations with clustering algorithms for completely unsupervised applications. Interestingly, it is also found that these compressed descriptors can still be competitive with the original ScSPM descriptors for classification tasks.

There are many popular feature learning techniques, and many variants of sparse coding in particular. Vector quantisation (VQ) and bag-of-words (BOW) techniques were popular before sparse coding [21, 41] and can be seen as a more restrictive instance of sparse coding (i.e. only using one dictionary element for encoding, instead of a small subset). Deep, or multiple-layer sparse architectures [11, 20, 67] are a very active area of research, and these architectures often achieve state of the art performance in classification tasks. Another popular area is kernel descriptors and multiple kernel learning for image recognition [18, 19, 47, 65]. The focus of this work will be on sparse single-layer[3] architectures, since they currently present the best trade off between performance and scalability with modest computational resources.

[3]It may be argued that the scale-invariant feature transform (SIFT) features used as inputs to these architectures in many works, including this, constitute an additional layer.

In Section 3.2 an overview of sparse coding is given, with emphasis on the algorithms used in this chapter. In Section 3.3 the general ScSPM framework for image representation is presented, and in Section 3.4 there is discussion about which dimensionality reduction methods are suitable for use with these high dimensional descriptors. In Section 3.5 experiments are carried out in order to test; (a) potentially more scalable replacements for sparse coding in the ScSPM framework, (b) How compressible these variants of ScSPM descriptors are for clustering and classification tasks, and (c) how generalisable various dictionaries are to alternate datasets for clustering and classification.

3.2 Sparse Coding Overview

The objective of sparse coding is to encode non-sparse signals, in this case image or SIFT patches $\mathbf{P} = [\mathbf{p}_1, \ldots, \mathbf{p}_N]^\top \in \mathbb{R}^{M \times N}$, as sparse linear combinations of elements in an over-complete dictionary, $\mathbf{D} = [\mathbf{d}_1, \ldots, \mathbf{d}_K] \in \mathbb{R}^{M \times K}$. These sparse combinations are referred to as the sparse codes, $\mathbf{C} = [\mathbf{c}_1, \ldots, \mathbf{c}_N]^\top \in \mathbb{R}^{K \times N}$. Learning the dictionary and codes is formulated as a regularised loss minimisation (or signal reconstruction) task,

$$\min_{\mathbf{D},\mathbf{C}} \sum_{n=1}^{N} \|\mathbf{p}_n - \mathbf{D}\mathbf{c}_n\|_2^2 \qquad \text{s.t.} \quad \|\mathbf{c}_n\|_0 \leq T \quad \forall n, \tag{3.1}$$

where $M < K$, and $T \ll M$, is a chosen number of non-zero elements in the sparse code. Also, $\|\cdot\|_2$ is the ℓ_2 or Euclidean norm, and $\|\cdot\|_0$ is the ℓ_0 norm[4]. Typically additional constraints, $\|\mathbf{c}_n\|_2 = 1$, and $\|\mathbf{d}_k\|_2 = 1 \; \forall n, k$ are enforced to avoid trivial solutions. Usually $\mathbf{D}$ is learned using just a subset of the signals, $\mathbf{p}_n$, and then fixed to encode the rest.

When $T = 1$ this is exactly the VQ problem, and when $\mathbf{c}_n$ is of unit length, this is K-means (KM), where $\mathbf{D}$ are the cluster centres, and $\mathbf{C}$ are the cluster assignments. When $T > 1$ this optimisation problem is combinatorially hard because of the ℓ_0

[4]This is the same as the number of non-zero elements in a vector.

constraint, in which the best of $\binom{K}{T}$ possible activation combination, or non-zeros, for $\mathbf{c}_n$ has to be chosen. Given $\mathbf{D}$, there are a number of greedy algorithms and heuristics which can approximately solve Equation 3.1 for $\mathbf{C}$. Some are iterative hard thresholding (IHT), orthogonal matching pursuit (OMP) [88], and the heuristic soft thresholding (ST) proposed in [33],

$$c_{k,n} = \max\{0, \mathbf{d}_k^\top \mathbf{p}_n - \alpha\} \quad \forall k, \tag{3.2}$$

where α is some threshold. There are also numerous methods for learning the dictionary. These algorithms typically alternate between optimising Equation 3.1 for $\mathbf{D}$ and approximating $\mathbf{C}$ with one of the greedy methods mentioned before. One such procedure is K-singular value decomposition (K-SVD) [2], which is a direct generalisation of K-means, and generally uses OMP for approximating $\mathbf{C}$.

A convex approximation to Equation 3.1, also called least absolute shrinkage and selection operator (LASSO) in regression analysis, is now synonymous with sparse coding (SC),

$$\min_{\mathbf{D},\mathbf{C}} \sum_{n=1}^{N} \|\mathbf{p}_n - \mathbf{D}\mathbf{c}_n\|_2^2 + \lambda \|\mathbf{c}_n\|_1 \quad \forall n, \tag{3.3}$$

again $\mathbf{d}_k$ and $\mathbf{c}_n$ are usually constrained to be unit length. This uses ℓ_1 regularisation[5] to enforce sparsity, which is tunable using a Lagrange multiplier, λ, sometimes chosen by cross-validation. The ℓ_1 regularisation naturally drives elements of the sparse code to zero, unlike an ℓ_2 norm, which prefers many small values, so is more suited to the loss/reconstruction objective of sparse coding. Many algorithms exist to solve Equation 3.3, see [6] for a summary. In this chapter the code from [128] is used for sparse coding, which uses the feature-sign sub-gradient method of Lee et al. [66]. Fully Bayesian solutions to Equation 3.3 also exist, which model the loss as a Gaussian distribution, and place a (non-conjugate) Laplace distribution prior on $\mathbf{c}_n$ [99] to induce sparsity. Similarly, a fully conjugate Beta process prior [86, 133] can be used to induce sparsity in the codes, though exact inference is still intractable. While these Bayesian methods usually perform very well in image reconstruction and

[5]Sum of the absolute values of the elements.

de-noising tasks they are usually non-convex so require potentially slower expectation propagation (EP), variational Bayes (VB) or even Markov chain Monte-Carlo (MCMC) sampling methods to learn.

In [121, 129] it was observed that in sparse coding, signals, $\mathbf{p}_n$, tend to be encoded by dictionary elements which are "local", or close in geodesic distance to the original signals. Yu et al. [129] showed this locality is an important factor in classification performance. In [121] the ℓ_1 regulariser in Equation 3.3 is replaced by a regulariser that enforces locality and naturally leads to sparsity (once thresholding has been applied),

$$\min_{\mathbf{D},\mathbf{C}} \sum_{n=1}^{N} \|\mathbf{p}_n - \mathbf{D}\mathbf{c}_n\|_2^2 + \lambda \|\mathbf{b}_n \circ \mathbf{c}_n\|_2^2 \quad \forall n. \tag{3.4}$$

Here $\circ$ is a Hadamard, or element-wise product, $\mathbf{b}_n = [k(\mathbf{p}_n, \mathbf{d}_1), \ldots, k(\mathbf{p}_n, \mathbf{d}_K)]^\top$ for a normalised Gaussian kernel $k(\cdot, \cdot) \in (0, 1]$, and each $\mathbf{c}_n$ is constrained to sum to 1. These are called locality-constrained linear codes (LLC). Interestingly this has an analytical solution, though in practice a fast approximation based on K-nearest neighbours (KNN) is used. These codes are shown by Wang et al. [121] to be comparable or better than the sparse codes used in [128] in terms of linear SVM classification performance (though they use a dictionary that is twice as large). They are also much faster to compute than optimising Equation 3.3 directly.

3.3 Image Coding Framework

It is not feasible to compute a single sparse code for each whole image in a large collection of images, $i \in \{1, \ldots, I\}$, so another representation for the image must be devised. A popular, and effective, representation is the ScSPM used by Yang et al. [128] which was adapted from the spatial pyramid match kernel used in [51, 65]. The representation for an image is as follows:

1. Extract a grid of overlapping patches from an image.

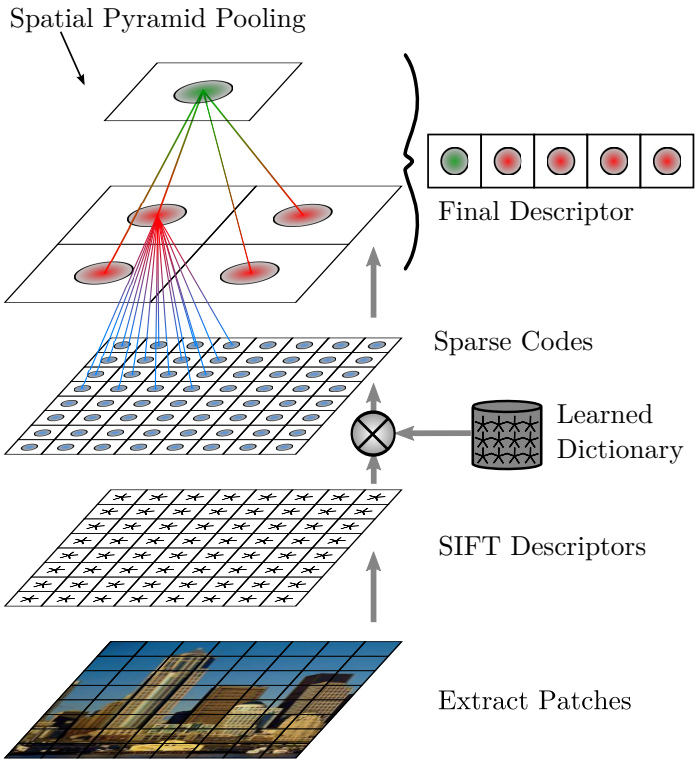


Figure 3.1 – The ScSPM pipeline for image descriptors, this configuration uses a $\{1, 2\}$ layer pyramid, typically a $\{1, 2, 4\}$ layer pyramid is used.

2. Encode each of these patches with a SIFT descriptor ($\mathbf{p}_n$), usually no keypoints are used.

3. Encode each of these SIFT patches as a sparse code ($\mathbf{c}_n$). A dictionary has been trained by this point from a random selection of SIFT descriptors.

4. Pool these sparse codes (usually using a max operator) in a spatial pyramid.

5. Concatenate all of the levels of the spatial pyramid into one descriptor for the image, $\mathbf{s}_i \in \mathbb{R}^D$, which is renormalised to unit length.

The image descriptors, $\mathbf{S} = \{\mathbf{s}_i\}_{i=1}^{I}$, are then used for classification, typically with a linear kernel SVM[6]. A graphical representation of this process is presented in Figure 3.1.

[6] $k(\mathbf{s}_i, \mathbf{s}_j) = \mathbf{s}_i^\top \mathbf{s}_j$

The spatial pyramid is essentially a hierarchy of image subdivisions over which to pool sparse codes. For example, a $l = \{1, 2, 4\}$ spatial pyramid pools sparse codes into $2^{l-1} \times 2^{l-1} \rightarrow \{1 \times 1, 2 \times 2, 4 \times 4\}$ image subdivision layers, starting with the finest-grained layer (4×4) first. The subsequent, less fine-grained layer is then a pooling of the pooled codes in the layer beneath it, and so on. These pooled codes are then concatenated into one, high dimensional, descriptor. This pyramid preserves some of the spatial layout in its description of the image, as opposed to the bag-of-features representations. It has been found that when using the *max* operator to pool sparse codes, as opposed to summing or averaging the codes, additional invariance to small affine transformations and clutter can be achieved [26, 60]. The max operator essentially preserves the maximum sparse code activation (dictionary-base response) in a pooling region, R, of the image (a large square in Figure 3.1),

$$\mathbf{c}_R = \left[\max_{n \in R} |c_{1,n}|, \ldots, \max_{n \in R} |c_{k,n}|, \ldots, \max_{n \in R} |c_{K,n}| \right]^\top. \tag{3.5}$$

It has been in shown by Boureau et al. [25, 26] that max-pooling may actually be a better statistical representation of sparse codes than average-pooling, as well as empirically having superior performance for classification.

No real justification for using SIFT descriptors is given in [128]. However in previous work [41] they have been shown to work better than coding raw image patches. Furthermore, multiple-layer sparse coding architectures, such as that in [20], have been required thus far to achieve similar performance on raw image patches without SIFT. It may be conjectured that the SIFT descriptors are serving a similar purpose as the first layer of these deep architectures [60].

Despite the spatial pyramid being an effective way to represent an image as a non-linear combination of its constituent sparse codes, it still leads to a very high dimensional descriptor. For example, if the dictionary has $K = 1024$ elements, and the three level $\{1, 2, 4\}$ spatial pyramid is used, then the final dimension for the image descriptor, $\mathbf{s}_i$, is $D = 1024 \times (1^2 + 2^2 + 4^2) = 21,504$!

3.4 Dimensionality Reduction for Image Descriptors

In many situations clustering algorithms are susceptible to the curse of dimensionality since they use distance metrics or similarity measures. However, it may be argued that this is less of a problem with sparse codes because "distant" codes usually lie on different subspaces of $\mathbb{R}^D$ [38]. Even so, clustering algorithms need to either iterate over *all* observations in a dataset, $\mathbf{S}$, or they require an $I \times I$ similarity matrix. When both I and D are large, even evaluating these similarities or distances can become prohibitively expensive, particularly for Mahalanobis based metrics. So it is desirable to find some mapping, $f : \mathbb{R}^D \rightarrow \mathbb{R}^d$ where $d \ll D$, so reduced dimensionality features, $\mathbf{x}_i \in \mathbb{R}^d$, can be clustered.

It is important for the low dimensional mapping, f, to preserve intra-cluster cohesion, and inter-cluster separation. If the observations lie on a low dimensional, smoothly varying, non-linear manifold embedded in high dimensional space, there are numerous algorithms that can uncover this structure. These algorithms also usually preserve local geodesic distances. Some examples are local linear embedding (LLE) [94], ISOMAP [111] and Laplacian Eigenmaps [10]. These algorithms are all computationally costly for very high dimensional data, i.e., 10,000 or so dimensions. They are also based on local connectivity graphs, which typically involve the computation of similarity matrices, so it is infeasible to apply them to many ScSPM descriptors. However, the success of ScSPM descriptors with linear classifiers suggests that linear dimensionality reduction techniques may be suited,

$$\mathbf{x}_i = f(\mathbf{s}_i) = \mathbf{U}\mathbf{s}_i, \tag{3.6}$$

where $\mathbf{U} \in \mathbb{R}^{d \times D}$ is a projection matrix. Linear techniques that attempt to preserve distances are some variants of multidimensional scaling (MDS) [37], and locality preserving projections (LPP) [55]. These methods also require local connectivity graphs or similarity matrices, so again it is potentially infeasible to apply these to large high

dimensional datasets.

This really leaves us with two options; PCA which does not explicitly preserve local distances, and random projections [7, 38]. Iterative PCA methods, such as the power and probabilistic methods introduced by Roweis [93], can be applied very successfully to very large and high dimensional datasets, and are used in Section 3.5. Random projection simply uses a projection matrix, $\mathbf{U}$, constructed by drawing each element independently from $\mathcal{N}(0,1)$, and then normalising $\mathbf{U}$ to have unit length rows. Interestingly, it has been proven to satisfy the Johnson-Lindenstrauss Lemma [8, 61],

Lemma 3.1 (Johnson-Lindenstrauss [61])**.** *Given a set of points, $\{\mathbf{s}_i\}_{i=1}^{I}$, where $\mathbf{s}_i \in \mathbb{R}^D$, there is a smooth (Lipschitz) function, $f : \mathbb{R}^D \to \mathbb{R}^d$ such that,*

$$(1-\epsilon)\|\mathbf{s}_i - \mathbf{s}_j\|_2^2 \leq \|f(\mathbf{s}_i) - f(\mathbf{s}_j)\|_2^2 \leq (1+\epsilon)\|\mathbf{s}_i - \mathbf{s}_j\|_2^2 \tag{3.7}$$

for all $i, j \in \{1, \ldots, I\}$, where $0 < \epsilon < 1$ and $d > \mathcal{O}(\log(D)/\epsilon^2)$.

So random projections preserve distances between codomains $\mathbb{R}^D$ and $\mathbb{R}^d$, with a proportional error, ϵ, so long as $d > \mathcal{O}\big(\log(D)/\epsilon^2\big)$. The aforementioned proof has also been extended to the case where $\mathbf{s}_i$ lie on some Riemannian manifold embedded in $\mathbb{R}^D$ [7]. In practice, this d may be much larger than when learned with an algorithm. There are also more recent methods which combine random projections with singular value decomposition (SVD) for very large matrix factorisation applications [54], though these are not used in this chapter.

In the experiments in Section 3.5 PCA and random projections will be used with the ScSPM descriptors for comparison in classification and clustering tasks. Random projections could also be used as prior compression stage to one of the other, more computation intensive, dimensionality reduction algorithms that preserve distance, such as was done by Gao et al. [46] and is suggested by [7].

3.5 Experiments

This section is concerned with quantifying how the following modifications to the ScSPM framework, for large scale applications, affect performance for classification and clustering tasks;

1. Are there alternative, more scalable, sparse coding techniques apart from the canonical sparse coding of Equation 3.3 that can be used in the ScSPM pipeline for classification and clustering?

2. Do the linear dimensionality reduction techniques such as PCA and random projections work for compressing ScSPM descriptors? How compressible are these descriptors? Will they work with clustering?

3. What alternative, more scalable, dictionary learning methods work well with the ScSPM pipeline? How generalisable are these dictionaries to novel data?

An attempt is made to use all of the original authors' code where possible, additionally the SIFT code used in [65] is used. For the classification results, the SVM with a linear kernel is used from [128], with the original parameters. 30 training images are used with 10-fold cross validation. Clustering is done with randomly initialised K-means (with ten replicates), given the true number of clusters. K-means is chosen because it is simple, and essentially the lowest common denominator in the clustering literature. Whitening is not performed in conjunction with PCA, since it was not found to improve clustering results for the ScSPM descriptors.

For validation, classification accuracy is quantified by the mean of the accuracies per class (or the diagonal of the confusion matrix) as is common in the literature. For clustering, three measures are used; the pair-counting based Rand index (RI) [91], the adjusted Rand index (ARI) [57], and the information theoretic normalised mutual information (NMI) [105], which is the same as the V-measure used in [92]. The presentation of the V-measure in [92] is intuitive because it is introduced as the harmonic mean of two opposing entropy based measures; *Homogeneity* and *Completeness*. A

cluster solution with a maximal level of homogeneity has data points that are members of a cluster comprised of only one single ground truth class. A cluster solution with a maximal level of completeness has all data points that are members of a single truth class belonging to a single cluster. All measures range from zero to one, with one being a perfect score (including RI and ARI). Homogeneity and Completeness are weighted equally in the V-measure for all of the experiments. These measures are used because they do not require each clustering solution to have the same number of clusters as the ground truth classes, and no manual reconciliation step is required – making for a fair comparison. Furthermore, it is sometimes useful to also analyse the components of V-measure to further tease apart clustering solutions. A good review of these metrics is presented in [118]. From here on it is implicit that NMI can equally refer to the V-measure.

All experiments use 40,000 random image patches to train their dictionaries, and a patch size of 16×16 pixels is used, with a stride of 8 pixels. The images are limited to have a maximum height or width of 300 pixels (aspect ratio is preserved). Max-pooling spatial pyramids with $\{1, 2, 4\}$ levels are used. These values are fairly common in the literature reviewed in Section 3.1 and Section 3.2, and refining these values is not in the scope of this chapter.

Three datasets are used in all of these experiments; the outdoor scenes dataset from [84], a ten class subset of the Caltech-101 dataset [42], and a novel dataset acquired by an autonomous underwater vehicle (AUV) from a deep reef off the East coast of Tasmania, Australia [125]. Exemplars of the classes within these datasets is shown in Figure 3.2.

The outdoor scenes dataset has eight classes which are; *coast, forest, highway, inside city, mountain, open country, street, tall buildings*. There are 2688 images in this dataset. A ten class subset of Caltech-101 is used with the classes; *airplanes, beaver, camera, cougar body, elephant, faces, laptop, leopard, motorbike, watch*. We only use a subset of the full 101 class dataset because it is unreasonable to expect K-means to maintain a good cluster-class correspondence with so many classes, and also runtime would be drastically increased, making a thorough empirical evaluation difficult.

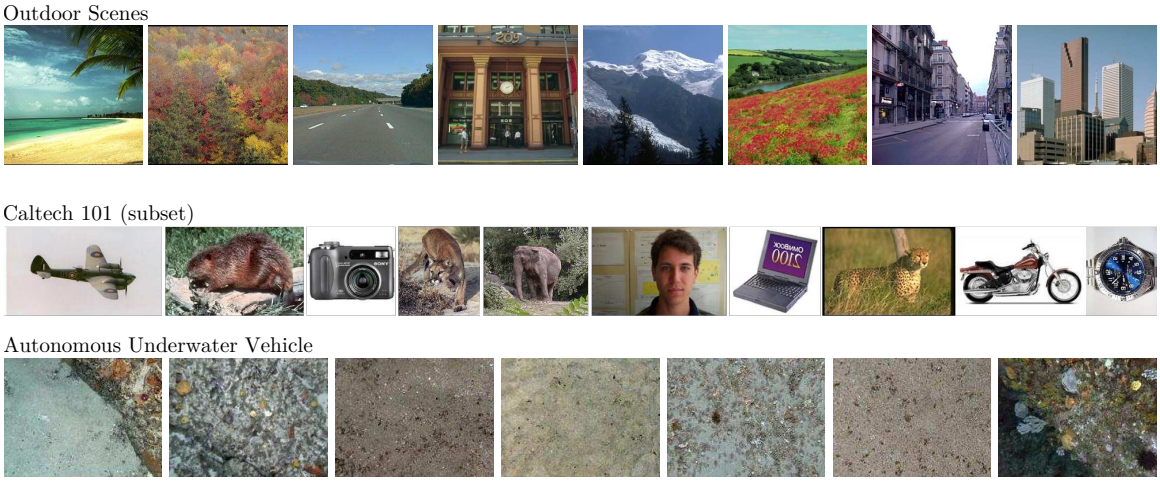


Figure 3.2 – Exemplar images of the three datasets used for comparison. The AUV dataset is a lot less visually diverse than the outdoor scenes and Caltech datasets. Please see the text for the class names.

This is also commonly done in the literature for similar reasons [50, 117]. There are 2760 images in this subset. The full dataset is used for some of the classification experiments however.

The AUV dataset is obtained from a downward facing stereo camera, though only the monochrome camera images are used. The dataset is of various sand and rocky reef habitats, which are in the photic zone, and are notionally taken at an altitude of two metres. There are 3035 images in this dataset, with seven classes; *sand/reef interface, low relief reef, coarse sand, fine sand, screw shell rubble, few screw shells, high relief reef*. The original dataset was twice as large, however there were considerable labelling errors in it. All of the obviously mislabelled images were removed. This dataset is a subset of those used in [100, 103].

All of the experiments were performed on a Core 2 Duo, 3.0 GHz machine with 4 GB of RAM. We do not present explicit run times as the code is from from disparate authors, and so cannot be fairly compared.

3.5.1 Comparing Sparse Encoders

In this section sparse encoders that are more scalable than SC are used in the ScSPM for classification and dimensionality reduction plus clustering. SC is also used as a benchmark. The following experiments were designed to clarify which encoders are suitable for large datasets without compromising performance. The following encoders are used;

ST with a threshold of $\alpha = 0.5$, as per [33]. This results in quite a large number of activations (non-zeros) for each code.

OMP with $T = 10$ activations. This was found to be a reasonable trade off in speed versus accuracy (less than this, performance reduced drastically, more than this, there were only small improvements). It also performed consistently well in [34]. The code from [2] is used.

SC with a regularisation setting of 0.3 as per [128], the code from this paper is also used. Apparently this also results in approximately $T \approx 10$ activations in each code.

LLC using 5 nearest neighbours (or elements) as per the recommendation of [121]. The original code from this paper is also used.

Even though the exact run times of these encoding methods cannot be compared, we can expect SC to be the slowest, having to solve a quadratic programming problem, and perform line search [66]. OMP is approximately $\mathcal{O}\big(T \times K\big)$ per iteration, when some pre-computation is performed [20]. LLC is $\mathcal{O}\big(K+nn^2\big)$ where nn are the number of nearest neighbours [121]. ST is simply vector multiplication with a max operation.

The experiments consist of running each encoder on each dataset using a K-means dictionary, $\mathbf{D}$, with 1024 elements trained on the corresponding dataset. This leads to the ScSPM descriptors having $D = 21,504$. SVM results are presented in Table 3.1, included are the clustering metrics for these results. Results are presented for K-means clustering for PCA ($d = 20$) compressed codes in Table 3.2, and randomly ($d = 1000$) compressed codes in Table 3.3.

Table 3.1 – SVM results on $\mathbb{R}^D$ features using a K-means dictionary (1024 elements). The figures in brackets denote one standard deviation.

Dataset	Enc.	Acc. (%)	RI	ARI	NMI/V
Caltech	ST	87.65 (1.67)	0.9804 (0.0039)	0.9445 (0.0113)	0.89 (0.0139)
	OMP	89.34 (1.64)	0.9868 (0.0028)	0.963 (0.008)	0.9214 (0.0114)
	SC	91.96 (1.24)	0.9888 (0.0039)	0.9687 (0.0111)	0.9332 (0.0123)
	LLC	88.68 (1.46)	0.9845 (0.0039)	0.9563 (0.011)	0.9120 (0.0116)
Outdoor	ST	77.99 (1.19)	0.9016 (0.0046)	0.5579 (0.0203)	0.6078 (0.0135)
	OMP	83.81 (0.88)	0.9239 (0.0029)	0.6593 (0.0126)	0.6998 (0.0102)
	SC	84.66 (0.83)	0.9274 (0.0034)	0.6733 (0.0151)	0.7056 (0.0107)
	LLC	82.73 (0.72)	0.9197 (0.0029)	0.639 (0.0134)	0.6746 (0.0124)
AUV	ST	59.84 (1.99)	0.7645 (0.0529)	0.4667 (0.1161)	0.4664 (0.0611)
	OMP	72.36 (1.1)	0.8511 (0.0071)	0.6581 (0.0187)	0.6177 (0.0144)
	SC	74.01 (0.93)	0.8628 (0.0158)	0.6875 (0.0411)	0.6371 (0.0215)
	LLC	75.46 (0.82)	0.861 (0.0104)	0.6815 (0.0268)	0.6409 (0.019)

We can see from Table 3.1 that SC is generally the best encoder for classification out of all of those used, with OMP close – usually within one standard deviation, then LLC closely behind that (though performing the best on the AUV dataset). ST does not perform as well as the other methods, which is not surprising.

The story is a little different in Table 3.2 and Table 3.3 with the clustering result. ST, and somewhat surprisingly LLC, consistently do badly relative to SC and OMP. For the PCA+K-means experiment, SC and OMP perform almost identically. However for the Random+K-means experiment, SC is quite consistently better than OMP.

To get more of a sense of why only some of the encoders work well with projection *and* clustering, we have plotted the first two principal components of each of the compressed code feature spaces in Figure 3.3. Although it is hard to judge the entire space from just the first two principal components, we can see that ST's projection does the worst job of separating the codes, followed by LLC. OMP and SC provide better separation, and are remarkably similar.

Based on the empirical evidence presented, it can be concluded that OMP is a suitable replacement for SC in the ScSPM pipeline for clustering and classification. Though a small trade off between performance and scalability has to be made. Interestingly, the experiments indicate that the locally constrained codes (LLC) do not work well with

linear dimensionality reduction and clustering. These experiments have not quantified how compressible all of these codes are, this is the subject of the next section.

3.5.2 Number of Dimensions to Preserve

The aim of this section is to quantify how compressible the ScSPM variants are, and how much performance is sacrificed for clustering and classification tasks with various levels of compression. Both PCA and random projections are used, and compression is used synonymously with dimensionality reduction in this context.

A subset of the experiments in the last section are repeated in this section, but for varying d. Additionally, the classification experiment in the last section is now also subject to prior dimensionality reduction to determine if similar performance can be achieved with compressed descriptors, $\mathbf{x}_i \in \mathbb{R}^d$. The results are presented in Figure 3.4. Only OMP sparse codes have been used for the clustering experiments for clarity.

We can see for all SVM experiments in Figure 3.4, the classification performance using the original ScSPM descriptors can be obtained using PCA-compressed descriptors when $d \approx 200$. Random projections also shows a similar capability, but requires $d > 2000$. We can see that the clustering results plateau well before the SVM results

Table 3.2 – PCA+K-means ($d = 20$) results for $\mathbb{R}^d$ features, K-means dictionary.

Dataset	Encoder	RI	ARI	NMI/V
Caltech	ST	0.8406 (0.0167)	0.4217 (0.0599)	0.6078 (0.0313)
	OMP	0.8579 (0.0161)	0.4829 (0.0635)	0.671 (0.0178)
	SC	0.8574 (0.0057)	0.4769 (0.0208)	0.6724 (0.0079)
	LLC	0.8473 (0.0052)	0.4399 (0.0218)	0.6132 (0.0150)
Outdoor	ST	0.8472 (0.0028)	0.339 (0.0072)	0.4419 (0.0109)
	OMP	0.8875 (0.0174)	0.5137 (0.0559)	0.6099 (0.0325)
	SC	0.8869 (0.0172)	0.5165 (0.0541)	0.6134 (0.0278)
	LLC	0.8577 (0.0082)	0.3784 (0.0210)	0.4867 (0.0169)
AUV	ST	0.7803 (0.0989)	0.5355 (0.1495)	0.4803 (0.0937)
	OMP	0.8161 (0.0184)	0.5415 (0.0556)	0.6068 (0.0282)
	SC	0.8153 (0.022)	0.5396 (0.0647)	0.5943 (0.037)
	LLC	0.7976 (0.0083)	0.4850 (0.0268)	0.5754 (0.0077)

Table 3.3 – Random+K-means ($d = 1000$) results for $\mathbb{R}^d$ features, K-means dictionary.

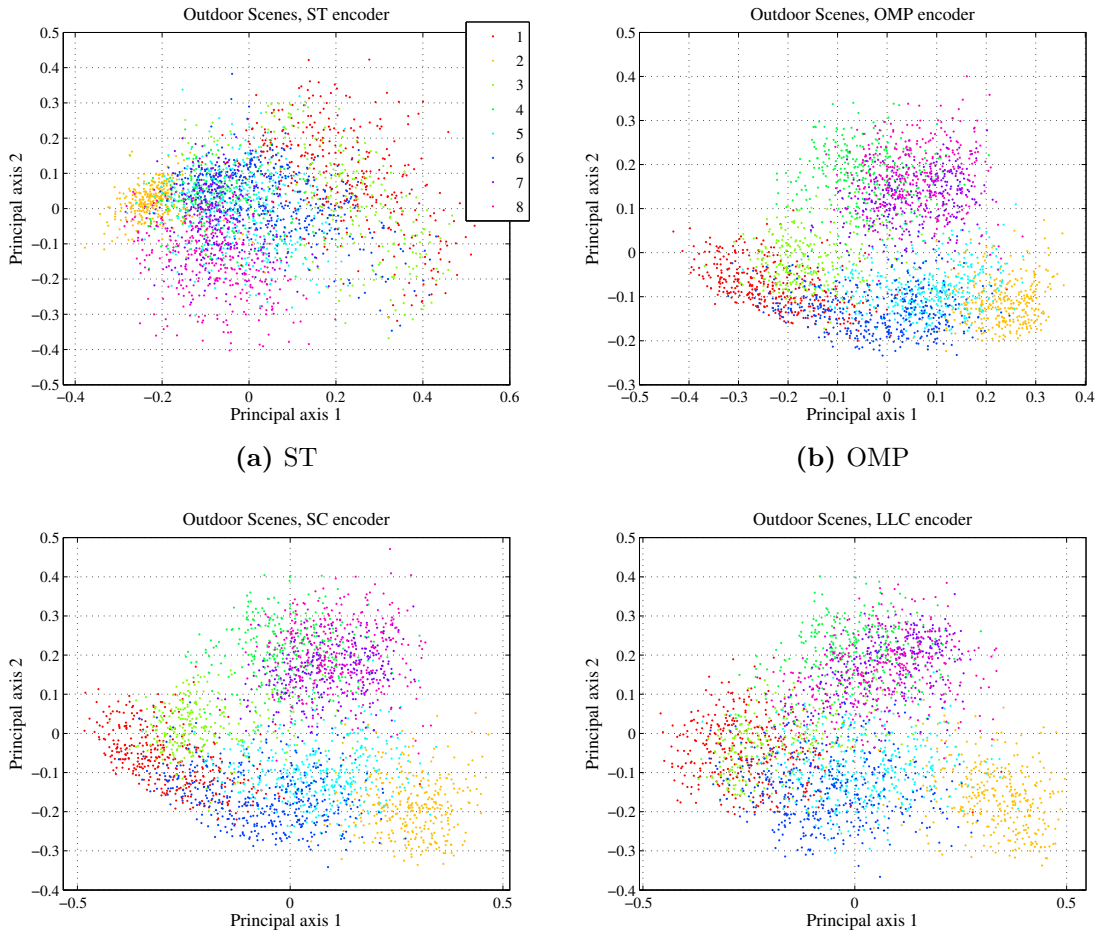
Dataset	Encoder	RI	ARI	NMI/V
Caltech	ST	0.8424 (0.0068)	0.4298 (0.0253)	0.6173 (0.0142)
	OMP	0.8523 (0.0089)	0.4575 (0.0317)	0.6532 (0.0259)
	SC	0.8628 (0.0209)	0.5104 (0.0753)	0.6727 (0.0317)
	LLC	0.8467 (0.0092)	0.439 (0.0334)	0.6143 (0.0257)
Outdoor	ST	0.8449 (0.0017)	0.332 (0.0065)	0.4384 (0.0104)
	OMP	0.8941 (0.0063)	0.53 (0.0223)	0.6062 (0.0145)
	SC	0.8956 (0.0091)	0.5416 (0.0346)	0.621 (0.021)
	LLC	0.8566 (0.0059)	0.37 (0.0226)	0.47689 (0.0236)
AUV	ST	0.6962 (0.158)	0.412 (0.2393)	0.4084 (0.1445)
	OMP	0.7892 (0.0116)	0.4639 (0.0378)	0.5455 (0.019)
	SC	0.8054 (0.0319)	0.5137 (0.0969)	0.5713 (0.0468)
	LLC	0.7921 (0.0125)	0.4657 (0.0406)	0.5583 (0.0170)

in Figure 3.4, typically at $d \approx 20$ for PCA, and $d \approx 1000$ for random projections. This is expected, since K-means is a far simpler algorithm than the SVM, and only operates on the original space, $\mathbb{R}^d$, and so may not be able to make full use of the high dimensions to separate descriptors. Whereas, SVMs map the data to a reproducing kernel Hilbert space, $\mathcal{H}$, to perform inference, in which the classes may be more separable, especially with the addition of training data.

We can also see that all of the clustering measures in Figure 3.4 are quite correlated, and so just the NMI will be used for the remainder of this thesis. It has also been noted in [118] that the NMI uses its range most effectively out of all of these clustering measures.

These experiments so far suggest that the ScSPM descriptors are highly compressible using PCA. This can be further seen in Figure 3.5, which shows the top 100 Eigenvalues of the projections for each dataset. They have been plotted in log-log space and appear very linear apart from the AUV codes, which initially decays faster before plateauing near zero. This suggests these Eigenvalues largely follow a power-law decay. Davenport et al. [38] states this is a very good indicator of a compressible signal.

The sparse codes of dissimilar signals, or classes of signals, tend to occur in different subspaces of $\mathbb{R}^K$, the space spanned by the sparse code dictionary [38, 121]. This is

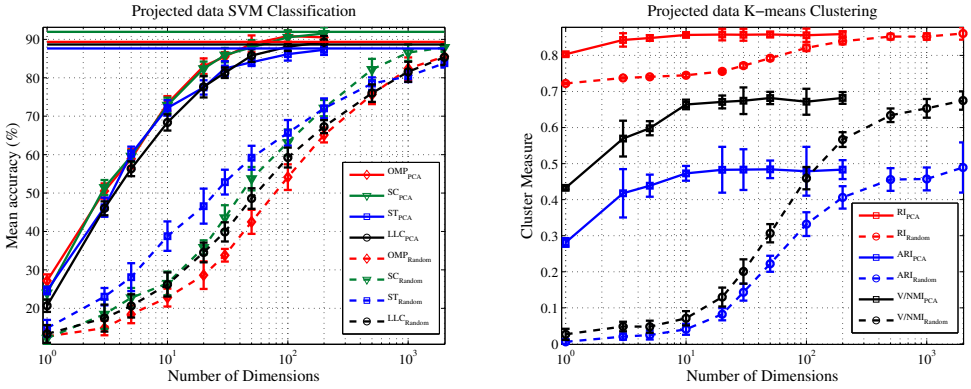


(a) ST **(b)** OMP

(c) SC **(d)** LLC

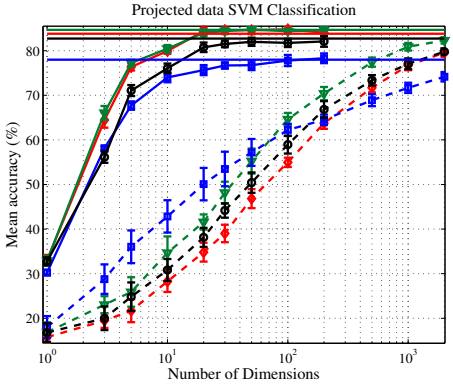
Figure 3.3 – First two principal axes of all of the encoders' feature spaces on the outdoor scenes dataset. It is interesting to note the similarity between that of OMP and SC.

because different signals tend to produce codes that have activations which correspond to different dictionary elements. This implies that as the number of classes increases in a dataset, the corresponding sparse codes will reside in more subspaces of $\mathbb{R}^K$, making them less compressible in unison. Thus d will have also have to increase to sufficiently distinguish these new classes. Formal verification of this conjecture is difficult because of the non-linear nature of the ScSPM pipeline, so it is empirically tested. The Caltech-101 classification experiment in [128] is replicated here, using *all* of Caltech-101, the original authors' code, and using 5-fold cross validation for

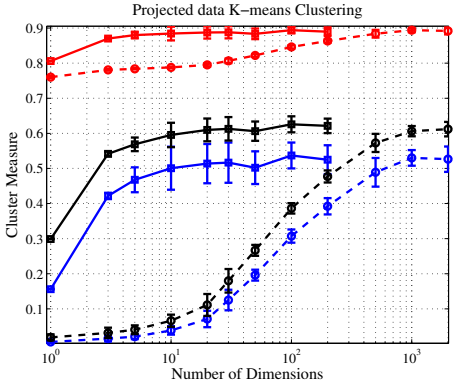


(a) Caltech-101, SVM

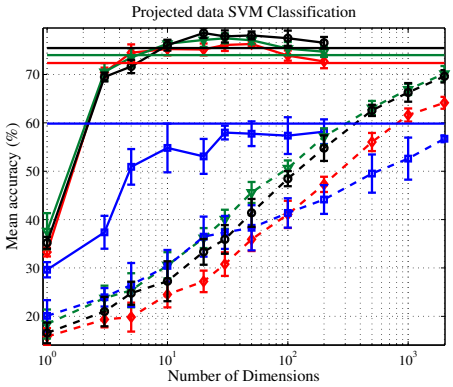
(b) Caltech-101, K-means



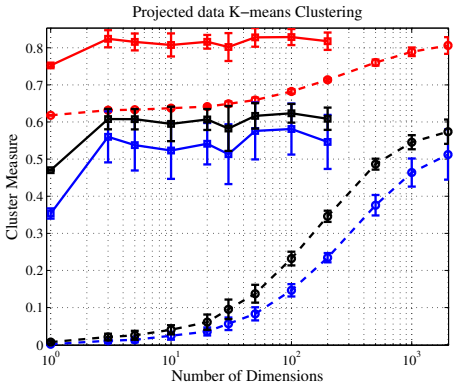
(c) Outdoor Scenes, SVM



(d) Outdoor Scenes, K-means



(e) AUV, SVM



(f) AUV, K-means

Figure 3.4 – Linear SVM classification and K-means clustering with the PCA and random projection compressed codes. The solid horizontal lines in the SVM plots are the mean accuracies using the original $\mathbb{R}^D$ codes. OMP codes have been used exclusively for the clustering results. K-means dictionaries with 1024 elements have been used.

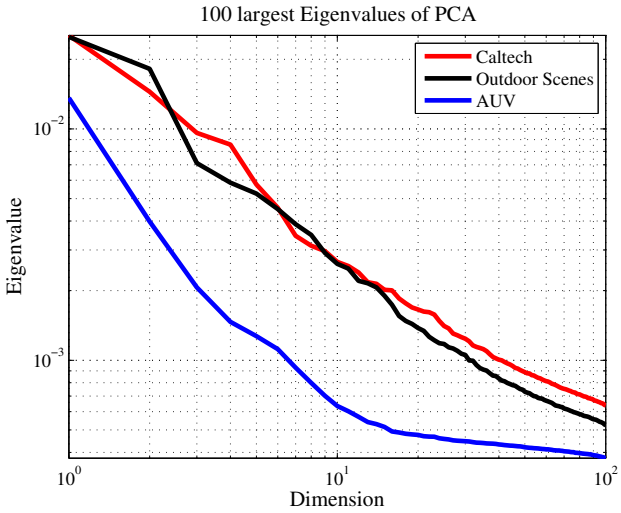


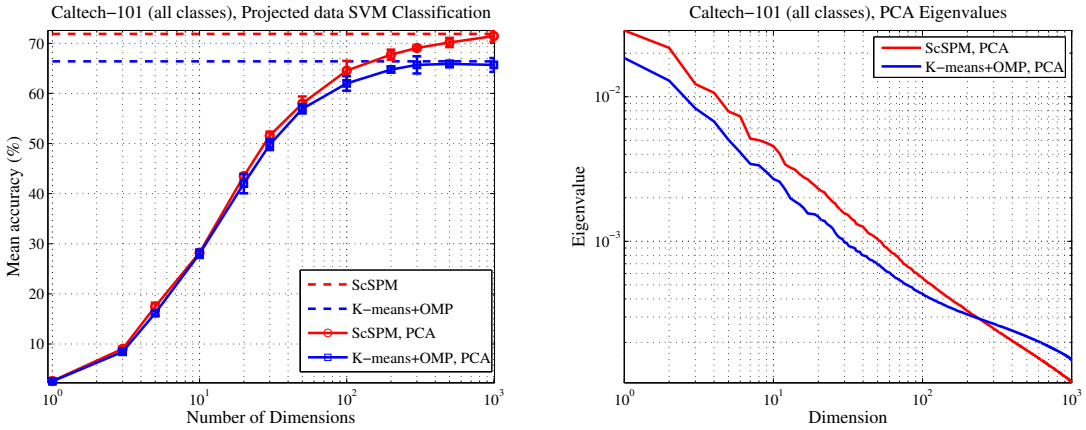
Figure 3.5 – The top 100 Eigenvalues of the PCA projections of the ScSPM descriptors using OMP encoding are shown using a log-log scale. With the exception of the AUV dataset, which initially decays faster, there is a linear trend, suggesting a power-law decay and highly compressible codes according to Davenport et al. [38].

the SVM. Here the ScSPM uses SC for both dictionary learning and encoding, and is compared to a K-means dictionary and OMP encoder. As before, the classification accuracies using the original ScSPM descriptors in $\mathbb{R}^D$ are compared with those PCA compressed descriptors, $\mathbb{R}^d$ for varying d. Results are presented in Figure 3.6.

Firstly from Figure 3.6 we can see that the original SC ScSPM performs better than the K-means+OMP ScSPM, which is to be expected from the previous results. Secondly, we can see that $d \approx 500$ to 1000 before the compressed descriptors approach the uncompressed descriptors accuracy, as opposed to the results in Figure 3.4. These descriptors are still massively compressible, but as conjectured, the number of classes present in the data dictates their compressibility. This experiment also exhibits Eigenvalues that have a power-law decay, suggesting that the spectral power (Eigenvalue sum) may be a good design criterion for choosing an appropriate dimensionality for $\mathbf{x}_i$.

3.5.3 Dictionary Comparison

Coates and Ng [33] and Boureau et al. [25, 26] present evidence that classification performance in the ScSPM framework is more dependent on the choice of sparse



(a) Mean Accuracy (%) **(b)** Eigenvalues (log-log scale)

Figure 3.6 – SVM classification using PCA compressed codes on all of the Caltech-101 classes. The original ScSPM experiment from [128] and OMP codes trained with a K-means dictionary are compared in (a). The dashed lines are the mean accuracies on the original (uncompressed) codes ($71.89 \pm 0.72\%$ using the original ScSPM pipeline). All eigenvalues of the projection are shown in (b). Here we can again see power-law decay.

encoder and max-pooling respectively, than the dictionary learning method. In this section it is also shown that dictionaries can generalise well across different datasets, as long as the dataset used to train the dictionary is sufficiently diverse in its appearance.

Four dictionary learning methods are tested in the ScSPM pipeline; K-means, SC using the code from [128], K-SVD also using the original authors' code [2], and simple random patches (RP), which was shown to work well in [33]. The random patches are simply randomly sampled SIFT descriptors, normalised to unit length. Both OMP and SC encoders are used, with the same settings from Section 3.5.1.

Presented in Table 3.4 and Table 3.5 are the SVM classification results of the original features in $\mathbb{R}^D$ for all of the dictionary learning methods with $K = 1024$, using OMP and SC coding respectively. The dictionaries are trained on the corresponding dataset at the beginning of the rows, and tested on all the other datasets across the columns. Additionally, a fourth dictionary training dataset is created consisting of an equal random contribution of SIFT descriptors from all datasets.

These results are mostly in agreement with [33], given a common encoder, each dic-

Table 3.4 – OMP+SVM cross-dataset dictionary comparison, Mean Accuracy (%). 1024 dictionary elements are used in all cases.

Dic. Dataset	Dictionary	Caltech	Outdoor	AUV
Caltech	RP	89.39 (2.14)	82.84 (0.49)	69.59 (1.57)
	KM	89.48 (1.73)	83.59 (0.83)	72.68 (1.15)
	SC	88.82 (1.40)	81.23 (0.79)	69.72 (1.74)
	K-SVD	89.78 (1.41)	83.21 (0.60)	70.28 (2.00)
Outdoor	RP	89.14 (2.10)	82.50 (0.65)	70.01 (1.63)
	KM	90.81 (1.44)	83.47 (0.99)	72.65 (1.12)
	SC	88.80 (1.50)	82.02 (0.65)	68.04 (1.05)
	K-SVD	89.74 (1.27)	83.51 (0.78)	71.73 (1.08)
AUV	RP	88.65 (1.55)	81.47 (0.49)	69.91 (1.54)
	KM	89.21 (1.29)	81.84 (0.65)	72.10 (1.44)
	SC	87.50 (1.81)	81.36 (1.11)	70.31 (1.35)
	K-SVD	89.27 (2.01)	81.84 (0.65)	71.78 (1.34)
Combination	RP	89.49 (1.32)	82.50 (0.88)	70.53 (1.20)
	KM	90.16 (2.23)	83.18 (0.80)	73.61 (1.14)
	SC	88.24 (1.73)	81.66 (0.98)	69.31 (1.85)
	K-SVD	89.07 (1.80)	83.01 (0.53)	71.91 (1.56)

Table 3.5 – SC+SVM cross-dataset dictionary comparison, Mean Accuracy (%). 1024 dictionary elements are used in all cases.

Dic. Dataset	Dictionary	Caltech	Outdoor	AUV
Caltech	RP	91.47 (1.58)	85.18 (0.80)	73.43 (1.84)
	KM	90.59 (1.67)	84.94 (0.87)	75.54 (1.22)
	SC	91.92 (0.93)	84.51 (1.08)	73.73 (1.33)
	K-SVD	91.36 (0.98)	83.96 (0.94)	74.32 (1.34)
Outdoor	RP	90.86 (1.55)	84.29 (0.58)	73.74 (1.49)
	KM	90.99 (1.36)	84.76 (0.77)	76.06 (0.95)
	SC	91.57 (1.43)	84.40 (0.83)	72.66 (1.65)
	K-SVD	91.46 (1.65)	84.88 (0.78)	74.68 (1.73)
AUV	RP	91.34 (1.37)	83.55 (0.64)	73.84 (1.60)
	KM	90.36 (1.23)	83.61 (0.37)	74.69 (1.07)
	SC	91.55 (1.39)	83.11 (0.75)	72.55 (1.48)
	K-SVD	91.71 (1.07)	83.93 (0.70)	74.23 (0.55)
Combination	RP	91.23 (1.31)	84.51 (0.74)	73.94 (1.08)
	KM	92.40 (2.18)	84.53 (0.68)	74.93 (1.43)
	SC	91.10 (1.33)	84.12 (0.63)	74.59 (1.19)
	K-SVD	91.08 (1.36)	84.57 (0.85)	75.35 (1.00)

tionary learning method yields similar results to other dictionary learning methods on the same dataset they were trained on. Most are within one standard deviation. If we look down the columns, we can see that this is also generally true of dictionaries trained on different datasets. The combination dataset also does not make significantly more or less generalisable dictionaries. SC does perform better than OMP on average. Some exceptions are; SC dictionaries seem to perform slightly worse with OMP encoding (in cases by slightly more than one standard deviation). There is also some slight reduction in accuracy when testing on the outdoor scenes dataset and using the AUV dataset to train the dictionary.

In Table 3.6 and Table 3.7 the same experiment is performed, but with K-means clustering on the PCA compressed features in $\mathbb{R}^d$ with $d = 20$. As established in the previous section, PCA was chosen as it tended to outperform random projections with fewer dimensions. Also this value of d was chosen because clustering performance had plateaued by this level in all of the datasets.

Like in Section 3.5.1, there is less of a clear difference between performance of OMP and SC codes when used for PCA and K-means clustering than for SVM classification. Implying less of a tradeoff is made between performance and scalability by using OMP in the ScSPM pipeline for clustering tasks. Initially it seems as if there may be more significant trends in these results compared to the SVM results. However, the standard deviation is relatively larger, and so it is hard to say with any confidence that any dictionary learning method generalises better than any other, even when trained on different datasets. The one exception again may be using AUV data to train dictionaries for the outdoor dataset, which tends to be worse.

This does not necessarily contradict the hypothesis that an encoder will perform as well on a dataset given a sufficiently diverse dictionary. As we can see from Figure 3.2, the images from the AUV dataset are a lot less diverse in appearance, than those in Caltech and the outdoor scenes datasets. This leads to the hypothesis that the AUV SIFT patches, $\mathbf{p}_n$, are not "tiling the space" [33] of the input data as much as the Caltech or outdoor scenes patches. This can be tested by looking at the Eigenvectors of the covariance of the patches, which is achieved with PCA. High

Table 3.6 – OMP+K-means cross-dataset dictionary comparison, NMI. 1024 dictionary elements are used in all cases.

Dic. Dataset	Dictionary	Caltech	Outdoor	AUV
Caltech	RP	0.6583 (0.0351)	0.5919 (0.0211)	0.5961 (0.0342)
	KM	0.6657 (0.0179)	0.6200 (0.0237)	0.6022 (0.0322)
	SC	0.6730 (0.0154)	0.5822 (0.0226)	0.6075 (0.0364)
	K-SVD	0.6936 (0.0121)	0.6087 (0.0172)	0.5877 (0.0270)
Outdoor	RP	0.6712 (0.0128)	0.5971 (0.0151)	0.6065 (0.0293)
	KM	0.6604 (0.0216)	0.6068 (0.0264)	0.5938 (0.0520)
	SC	0.6711 (0.0148)	0.6126 (0.0158)	0.5711 (0.0356)
	K-SVD	0.6685 (0.0162)	0.6144 (0.0174)	0.5734 (0.0414)
AUV	RP	0.6893 (0.0073)	0.5503 (0.0130)	0.5930 (0.0313)
	KM	0.6783 (0.0039)	0.5740 (0.0226)	0.6035 (0.0177)
	SC	0.6687 (0.0094)	0.5925 (0.0131)	0.5933 (0.0348)
	K-SVD	0.6928 (0.0134)	0.5688 (0.0211)	0.5891 (0.0389)
Combination	RP	0.6673 (0.0151)	0.5962 (0.0160)	0.5882 (0.0544)
	KM	0.6590 (0.0128)	0.6138 (0.0389)	0.5752 (0.0464)
	SC	0.6722 (0.0109)	0.5740 (0.0199)	0.5885 (0.0341)
	K-SVD	0.6695 (0.0196)	0.6063 (0.0161)	0.5845 (0.0327)

Table 3.7 – SC+K-means cross-dataset dictionary comparison, NMI. 1024 dictionary elements are used in all cases.

Dic. Dataset	Dictionary	Caltech	Outdoor	AUV
Caltech	RP	0.6681 (0.0071)	0.6335 (0.0216)	0.6147 (0.0338)
	KM	0.6723 (0.0156)	0.6382 (0.0218)	0.5837 (0.0611)
	SC	0.6804 (0.0102)	0.6005 (0.0181)	0.5883 (0.0249)
	K-SVD	0.6449 (0.0195)	0.5794 (0.0113)	0.6159 (0.0424)
Outdoor	RP	0.6727 (0.0110)	0.6159 (0.0224)	0.5899 (0.0277)
	KM	0.6527 (0.0242)	0.6173 (0.0293)	0.5855 (0.0312)
	SC	0.6700 (0.0146)	0.6096 (0.0161)	0.5928 (0.0414)
	K-SVD	0.6689 (0.0296)	0.6121 (0.0211)	0.6235 (0.0326)
AUV	RP	0.6765 (0.0103)	0.5916 (0.0150)	0.5939 (0.0397)
	KM	0.6648 (0.0065)	0.5859 (0.0220)	0.5774 (0.0198)
	SC	0.6717 (0.0063)	0.5863 (0.0182)	0.5926 (0.0421)
	K-SVD	0.6738 (0.0123)	0.5967 (0.0205)	0.5834 (0.0437)
Combination	RP	0.6719 (0.0107)	0.6223 (0.0225)	0.5836 (0.0529)
	KM	0.6752 (0.0139)	0.6300 (0.0210)	0.6079 (0.0499)
	SC	0.6717 (0.0264)	0.6166 (0.0170)	0.5960 (0.0374)
	K-SVD	0.6725 (0.0133)	0.6113 (0.0214)	0.5831 (0.0232)

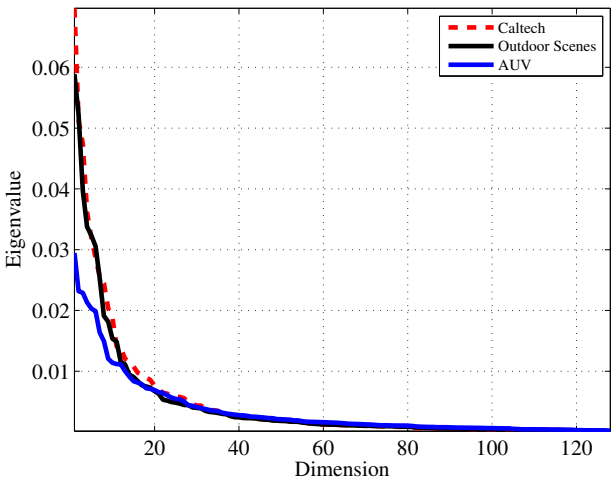


Figure 3.7 – The Eigenvalues of 40,000 random SIFT image patch descriptors. These are *not* in log-log space.

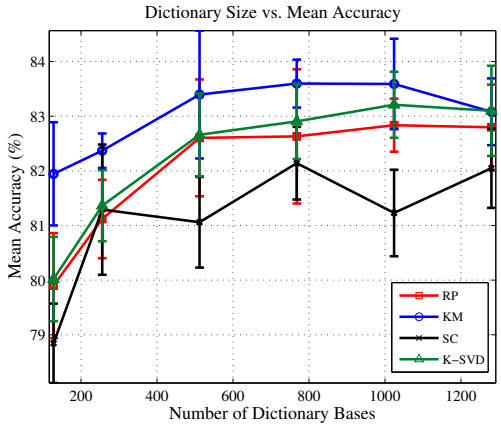
magnitude Eigenvectors are directly related to having a large variance, or extent, in each principal dimension[7] (inclusive of outliers). The Eigenvectors for 40,000 random SIFT patches for the three datasets are plotted in Figure 3.7. The SIFT descriptors have been normalized as per Lowe [75], and so should all have a similar scale.

We can see from Figure 3.7 that, while the same space is spanned by all datasets (there are no zero Eigenvalues), the AUV SIFT patch Eigenvalues are generally much lower, especially in the higher variance dimensions.

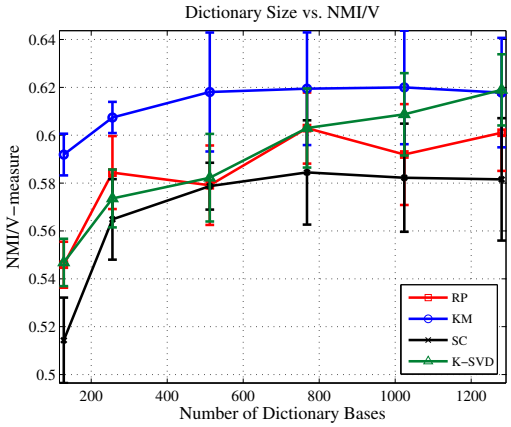
An exhaustive study was also performed on how these results generalise with varying the number of dictionary elements, K. In the interest of keeping the experimental state-space presented to the reader manageable, only a few results which exemplify the study are presented. The Caltech subset, outdoor scenes, and AUV dictionaries were applied to classification and clustering the outdoor scenes datasets in Figure 3.8. OMP codes, and values for K of 128, 256, 512, 768, 1024 and 1280 were used for these experiments.

We can see that there is compelling evidence for K-means performing well on this

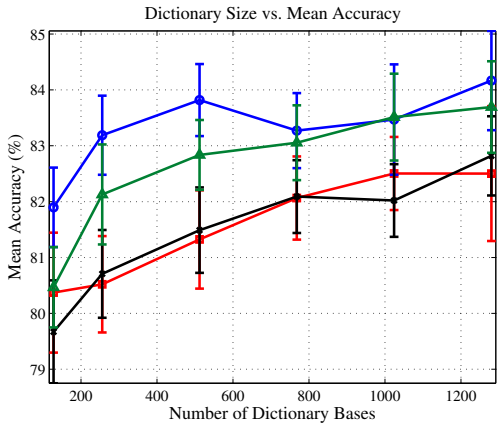
[7]There are other ways to test the ability of a dictionary to reconstruct sparse signals, such as the *mutual coherence* of a matrix [38]. However it has been suggested by [2] that, practically, this measure may be too pessimistic for classification tasks.



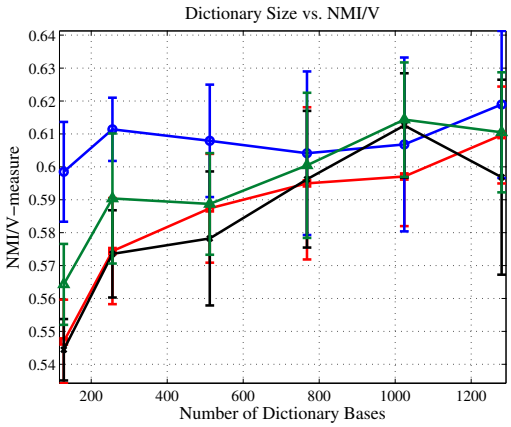
(a) SVM, Caltech



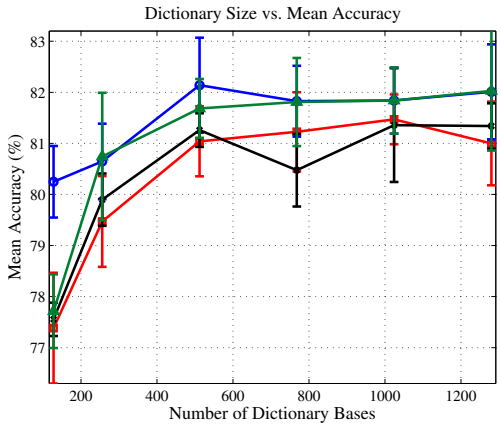
(b) PCA + K-means, Caltech



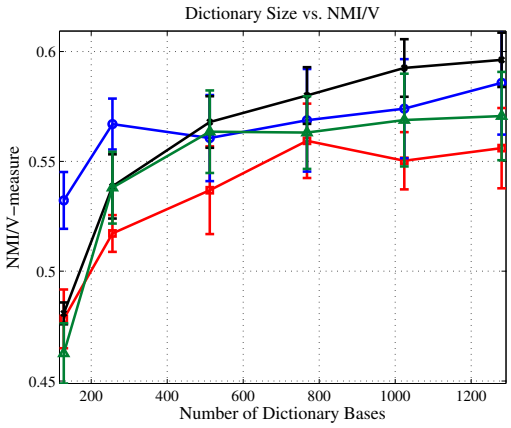
(c) SVM, outdoor scenes



(d) PCA + K-means, outdoor scenes



(e) SVM, AUV



(f) PCA + K-means, AUV

Figure 3.8 – SVM and K-means performance vs. dictionary size on the *outdoor scenes* dataset using OMP encoding. The datasets used to train the dictionaries are given in the sub figure captions.

dataset. This does not imply it will perform well on other datasets, but it does imply that specific dictionary algorithms may work best for particular datasets, despite the conclusions drawn in [33]. Again we see that the AUV dataset dictionaries are not generalisable to the outdoor scenes datasets, re-enforcing the caveat that the dictionary training dataset must exhibit enough variation to effectively "tile the space" of the dataset to be encoded. While performance dropped off for smaller dictionary sizes in all cases, the generalisability of dictionaries relative to the dictionaries trained on the original datasets seemed to be consistent. That is, the performance did not decrease towards the y-axis at a rate substantially faster than the dictionaries trained on the original datasets.

3.6 Summary

In this chapter it was demonstrated that ScSPM descriptors can be compressed by many orders of magnitude, while still maintaining the same classification accuracies as the original codes with a linear SVM. Furthermore PCA appears to perform as well as Random Projections, with far fewer dimensions, despite the fact there are no guarantees that PCA preserves pair-wise distances.

It was also shown that similar performance can be achieved for K-means clustering tasks using both SC and OMP in the compressed ScSPM pipeline. Both outperform ST and LLC for these tasks. OMP is a much faster algorithm, so it is a viable alternative to SC for large datasets. However, SC still markedly outperforms OMP both before and after projection when used for classification.

The results of Coates and Ng [33] have somewhat been replicated in that it was observed the classification accuracy is more affected by the encoder in the ScSPM framework, rather than the dictionary type. This assumes the dictionary used tiles the space of image patches sufficiently. This appears to also hold for projection and clustering. However, when dictionary size is taken into account there does appear to be evidence for certain dictionary learning methods to slightly favour specific datasets.

This result was extended to show that classification and clustering performance can be maintained even when dictionaries are trained *across* datasets. So long as training dataset is sufficiently diverse in appearance, the learned dictionary can generalise to new datasets well for classification and clustering tasks. It was observed that dictionaries trained on the AUV dataset did not perform as well as dictionaries trained on the other datasets. This can be satisfactorily explained by the AUV dataset not having diverse imagery compared to the other datasets. This was examined by observing the magnitude of the Eigenvectors of the covariance for each of the datasets.

The astute reader may also notice that while the ScSPM descriptors preserve the crude spatial layout of the images, the AUV images are quite unstructured, and the same scene may be captured multiple times from rotated vehicle poses. A pyramid of only one layer was tried with the AUV images to enforce rotational invariance. However, it was found that performance was consistently lower for classification and clustering tasks. This may be because the additional pooling layers are capturing other important structural features in the AUV imagery. It is also conceivable that the dimensionality reduction applied to the ScSPM descriptors learns a rotational invariance for this dataset (explaining the increased performance over the non-compressed descriptors in Figure 3.4), however further analysis would need to be conducted to verify this.

In general, from these results it is recommended that the original ScSPM framework be used for small datasets, or where performance is paramount. Although, a faster dictionary learning method such as K-SVD could be used in place of SC without sacrificing performance greatly. For large datasets where it may not be tractable to even use SC to encode patches for ScSPM descriptors, the experiments presented lead to the following recommendations:

- OMP can achieve encoding performance close to that of SC, especially for clustering tasks after compression, and is more scalable.
- PCA can be used very effectively with the ScSPM framework for both SC and OMP codes. This allows clustering algorithms to used these descriptors, and

potentially allows SVMs to use more training data, and non-linear kernels.

- A dictionary does not necessarily have to be learned for each new dataset, or relearned for new data, so long as the data used to train the dictionary was sufficiently diverse in appearance. This may be useful for incremental classification/clustering applications.

- In Figure 3.6 a relationship between the number of latent classes, and compressibility of the ScSPM descriptors was observed. However, the PCA Eigenvectors of the ScSPM descriptors can give insight into how to choose a suitable d.

These results allow modified and compressed ScSPM descriptors to be used for clustering large datasets (100,000 images) in the next chapter, without the need for training a dictionary specially for the task. This would not be feasible with the original ScSPM framework of [128].

As future work, it would be useful to see if there exists a way to combine sparse coding, spatial pyramid pooling, and dimensionality reduction into a single algorithm. This has been done for sparse coding and dimensionality reduction by Gkioulekas and Zickler [49], but it does not take into account the spatial pyramid pooling stage, which also may not be tractable. Also, it is this author's experience that the filters learned in [49] look visually very similar to the pseudo-inverse filters learned by independent component analysis (ICA) [58]. It would also be interesting to see if there is a similarity between these two algorithms.

Chapter 4

Clustering Groups of Related Visual Datasets

Large image collections are frequently partitioned into distinct but related groups, such as photo albums from similar environments that contain similar scenes. From a clustering point of view, these groups (e.g. albums) may share clusters (e.g. scenes), with proportions that are specific to the group. These group-specific cluster proportions may be thought of as a type of "context" for the image clusters. In this chapter, an effective hierarchical Bayesian model for clustering this type of data is presented. It uses a deterministic variational Bayes algorithm for learning and to choose the number of clusters that are shared across groups. A model is formulated that outperforms more conventional clustering models for this novel task (in both performance and runtime). The main contribution is in providing evidence that uncovers why performance is improved for this type of model. It is tested on standard computer vision datasets, a large dataset of underwater stereo imagery collected from multiple autonomous underwater vehicle (AUV) surveys, and a collection of holiday photo albums.

4.1 Introduction

Supervised classification approaches have demonstrated great performance for object and scene recognition in visual datasets [41, 72, 128]. For very large datasets with many classes, producing the training data can represent a substantial, and potentially expensive, human effort. In these situations there is scope for the use of unsupervised clustering approaches that can discover labels automatically. Even if the labels found do not have the precise semantic meaning of classified imagery, there is value in the rapid data summaries that are produced. The motivating example comes from the problem of labelling large visual datasets of the seafloor obtained by an AUV for ecological analysis. It is expensive to label this data, as taxonomical experts for the specific region are required. Quick, approximate summaries of quasi-habitats can be generated by unsupervised methods "for free" (i.e. no effort on behalf of the expert). These can be used to focus the efforts of experts, and inform decisions on additional sampling.

Many clustering algorithms such as K-means, mixture models, and spectral methods [81], assume that data comes from a single dataset or group. There are situations where multiple datasets, or groups, have observations that are statistically related but vary slightly. For example, photos in albums from various events or holidays may have similar scenes, such as mountains, beaches, parties, etc., but the make-up of each album will vary depending on the featuring events. Similarly, multiple AUV surveys are usually conducted in a region. A-priori it can be assumed that the surveys contain images of similar habitats, but the proportion of those habitats appearing in each survey will differ. Rather than concatenating this data, or clustering the data as completely disjoint sets, it is shown that it desirable to share statistical strength between groups via cluster parameters, while still keeping the proportions of clusters in the groups distinct. Essentially modelling the "context" in which the images occur.

A hierarchical Bayesian mixture model is presented that can take advantage of this group structure when clustering. It is similar in structure to latent Dirichlet allocation (LDA) [17], and is referred to as the grouped mixtures clustering model (GMC). The

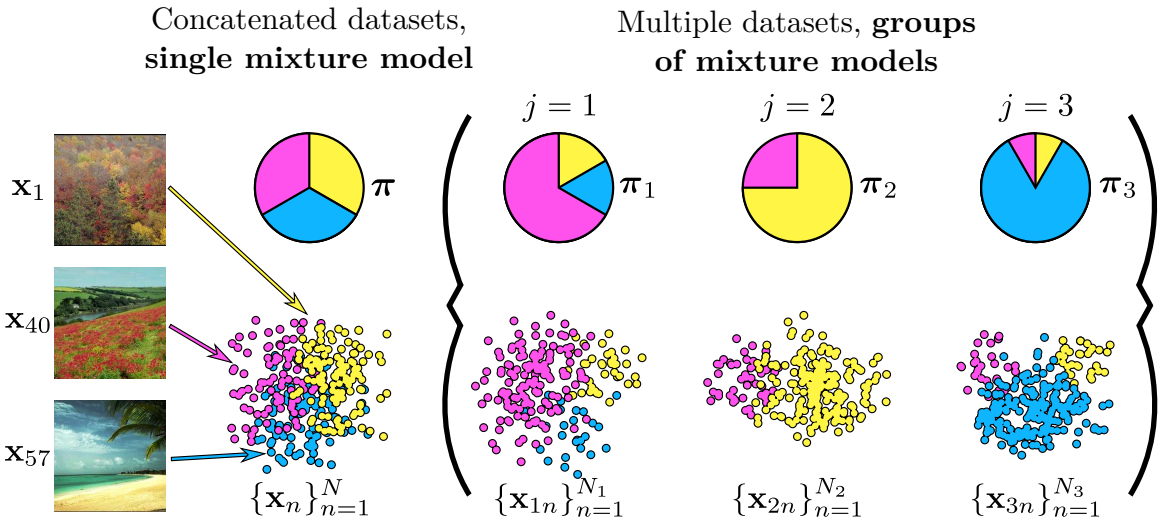


Figure 4.1 – Demonstration of jointly clustering datasets (groups). On the right are three related datasets or albums, $j = \{1, 2, 3\}$, which have the same clusters (image scene types) within them. Each image, $\mathbf{x}_{jn}$, is a point. Each group has a different proportion, $\boldsymbol{\pi}_j$, of the clusters indicated by the pie charts. The group mixture model presented in this chapter models the structure of this data. On the left is a concatenation of these groups into one meta-dataset. A single mixture model can be used to model this data structure, but as can be seen, it is harder to disambiguate the clusters from only this one view of the observations in feature space.

contribution of this work is to show that by exploiting the structure of this data the model can obtain better, more homogeneous, clusters than similar models that do not distinguish between the groups. It is also shown that inference takes less time when using this model for large datasets. The presented experiments show that by keeping the proportions of clusters in each group distinct, the learning algorithm can more easily disambiguate between clusters that occur close in feature space, but may not co-occur in the same frequency (or at all) within each group. Essentially, by preserving the structure of the groups, novel views of observations in feature space can be used by the algorithm to help separate highly overlapping clusters, as illustrated by Figure 4.1.

This problem is similar to *multi-task* and *ensemble* clustering [53, 105, 131]. The aim for these problems is to find relationships between distinct clustering solutions for different types of data or, more commonly, for each run of a randomly initialised

clustering algorithm on one dataset. These relationships are then used to enhance the clustering results by, for example, proposing that two clusters across two clustering results are similar. This usually proceeds in an iterative fashion, and has been shown to improve the clustering solutions. The approach here is different because it is assumed from the outset the same clusters are shared between different groups of the same type of data. So, the problem is to discover these clusters inherent in multiple datasets in a data driven fashion using a single run of one algorithm.

Inspiration for this work has been taken from the field of information retrieval for text corpora, and models such as LDA [17], and the non-parametric Bayesian hierarchical Dirichlet process (HDP) [109]. These models also exploit the structure inherent in their datasets to enhance inference; words in documents are analogous to observations in groups. They are used for finding latent factors, or topics, in discrete data, and can be seen as part of a larger family of clustering and dimensionality reduction techniques [17, 29].

Hierarchical Bayesian models, such as the aforementioned topic models, have been used extensively in the computer vision literature for supervised bag-of-words based object recognition and segmentation [42, 78], and scene classification [21, 24, 41]. Similar models have also been used for unsupervised object detection and image segmentation [39, 107, 108]. Mixture models have previously been applied to clustering whole images in [50, 79, 103], which is the application closest to the one presented here. However, to the author's knowledge, clustering data in higher-level groups such as albums is a relevant and entirely novel application in the computer vision literature, to which no algorithms have yet been applied.

In the next section the GMC is presented, as well as a discussion of its structure in light of the application. In Section 4.3 an algorithm for learning the GMC and choosing the number of clusters using variational Bayes is presented. For comparison, symmetric-prior, and fast but less accurate "sparse" variants of the GMC are introduced in Section 4.4, and the image representation is discussed in Section 4.5. In Section 4.6 the GMC and other clustering algorithms are compared on the 8-class outdoor scenes dataset from [84], a subset of the Caltech-101 object classes dataset [42], a large

visual dataset obtained from multiple AUV surveys of a reef environment, and finally a novel dataset comprising photos from holiday albums.

4.2 Sharing Clusters Between Groups

In this section the generative GMC model is presented to solve the problem depicted in Figure 4.1. The GMC resembles smoothed LDA [17], and shares the idea that observations (words) share specific contextual information provided by their group (document). However a document in this case is a *whole album*, and a word is analogous to an *image*. This is in contrast to LDA as used by Fei-Fei and Perona [41] where documents are images, and words are image parts (quantised scale-invariant feature transform (SIFT) descriptors). This model is not referred to as LDA, mainly to distinguish its application, and also because the distributions used are different.

It is assumed that mixture weights are random draws from a *generalised Dirichlet* distribution [36], $\mathrm{GDir}(a_1, \ldots, a_{K-1}, b_1, \ldots, b_{K-1})$. The generalised Dirichlet includes the Dirichlet distribution as a special case when $b_k = a_{k+1}+b_{k+1}$ for $k \in \{1, \ldots, K-2\}$. One reason for using the generalised Dirichlet distribution is its similarity to a Dirichlet process [59], which is the non-parametric extension of a Dirichlet distribution. This has been shown to perform better than a Dirichlet prior for text modelling applications [109]. An attempt was made to apply a HDP to this problem, however it either converged poorly when a conjugate representation was used (because of complex interactions between latent indicator variables), or is quite algorithmically complex when using more accurate collapsed forms [110]. The GMC is a simple, parametric alternative to a one level HDP for this problem. Similarly, there is evidence in the text modelling literature of asymmetric priors on weights performing better than symmetric Dirichlet priors [120]. The generalised Dirichlet is quite easily made asymmetric with simple choices for its parameters, as done in Section 4.6.

Following from Figure 4.1 observations, or images $\mathbf{x}_{jn} \in \mathbb{R}^D$, are assumed to be arranged in the following manner:

- There are N_j images in a group, or "album", $\mathbf{X}_j = \{\mathbf{x}_{jn}\}_{n=1}^{N_j}$.
- There are J groups, or albums, in the whole dataset, $\mathbf{X} = \{\mathbf{X}_j\}_{j=1}^{J}$.

The aim is to discover K clusters, or mixture components, common to these groups with parameters $\boldsymbol{\Theta} = \{\theta_k\}_{k=1}^{K}$. The jth group is described by the proportions of the image clusters within it, $\boldsymbol{\pi}_j = [\pi_{j1}, \ldots, \pi_{jk}, \ldots, \pi_{jK}]$, where $\pi_{jk} \in [0, 1]$ and $\sum_k \pi_{jk} = 1$. Latent labels, $\mathbf{Z} = \{\mathbf{z}_j\}_{j=1}^{J}$, are used as auxiliary variables to assign observations to the clusters. The GMC has the following generative process once all of the cluster parameters have been drawn, $\theta_k \sim p(\eta, \boldsymbol{\nu})\, \forall k$. For group j:

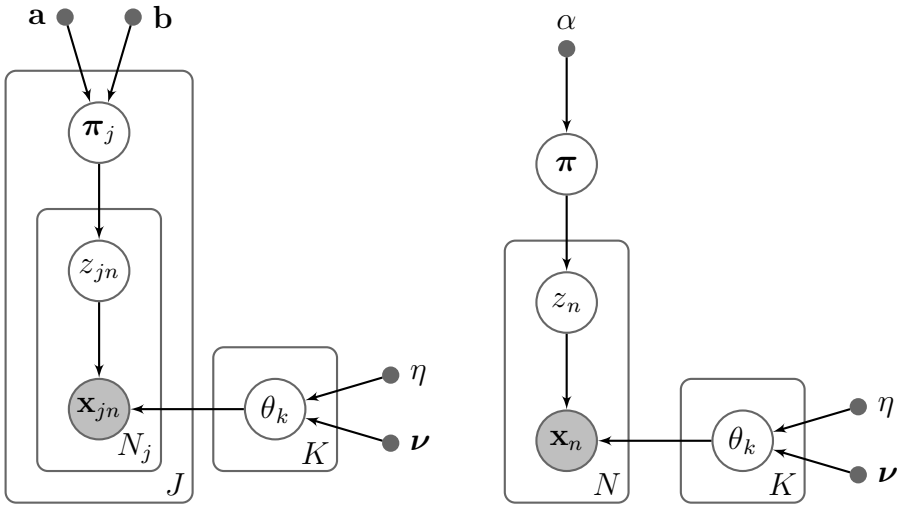
1. Draw group mixture weights, $\boldsymbol{\pi}_j \sim \mathrm{GDir}(\mathbf{a}, \mathbf{b})$.
2. For each of the N_j observations in group j,
   (a) Choose a cluster, $z_{jn} \sim \mathrm{Cat}(\boldsymbol{\pi}_j)$, where $z_{jn} \in \{1, \ldots, K\}$.
   (b) Draw an observation, $\mathbf{x}_{jn} \sim p(z_{jn}, \boldsymbol{\Theta})$, from an exponential family distribution with parameters $\boldsymbol{\Theta}$, conditioned on the corresponding label, z_{jn}.

The collection of all group mixture weights is termed $\boldsymbol{\Pi} = \{\boldsymbol{\pi}_j\}_{j=1}^{J}$. The graphical model of the GMC is presented in Figure 4.2, and the corresponding joint distribution is,

$$p(\mathbf{X}, \mathbf{Z}, \boldsymbol{\Pi}, \boldsymbol{\Theta} | \mathbf{a}, \mathbf{b}, \eta, \boldsymbol{\nu}) = \prod_{k=1}^{K} p(\theta_k | \eta, \boldsymbol{\nu}) \times \prod_{j=1}^{J} \mathrm{GDir}(\boldsymbol{\pi}_j | \mathbf{a}, \mathbf{b}) \prod_{n=1}^{N_j} \mathrm{Cat}(z_{jn} | \boldsymbol{\pi}_j)\, p(\mathbf{x}_{jn} | z_{jn}, \boldsymbol{\Theta})\,. \quad (4.1)$$

In this joint, the $p(\mathbf{x}_{jn} | z_{jn}, \boldsymbol{\Theta})$ further factorises,

$$p(\mathbf{x}_{jn} | z_{jn}, \boldsymbol{\Theta}) = \prod_{k=1}^{K} p(\mathbf{x}_{jn} | \theta_k)^{\mathbb{1}[z_{jn}=k]}\,, \quad (4.2)$$



(a) GMC **(b)** Bayesian mixture model

Figure 4.2 – Graphical model of the GMC (a), and a regular Bayesian mixture model (b) with a symmetric Dir(α) prior. The dots indicate point estimates of the hyper-parameters, the shaded nodes $\mathbf{x}_{jn}$ and $\mathbf{x}_n$ are observable, and the plates denote replication over the index in their lower right corner.

where $\mathbf{1}[\cdot]$ is an indicator function that returns 1 when the condition in the brackets is true, and 0 otherwise. The generalised Dirichlet prior on the group mixture weights, $\text{GDir}(\boldsymbol{\pi}_j|\mathbf{a},\mathbf{b})$, is essentially the same as a truncated stick-breaking process [13, 59],

$$\pi_{jk} = v_{jk}\prod_{l=1}^{k-1}(1-v_{jl}), \qquad v_{jk} \sim \begin{cases} \text{Beta}(a_k, b_k) & \text{if } k < K \\ 1 & \text{if } k = K, \end{cases} \tag{4.3}$$

where $v_{jk} \in [0,1]$ are "stick-lengths" for each group, and $\text{Beta}(\cdot)$ is a Beta distribution. It must be stressed that the generalised Dirichlet is used as a prior placed over the weights, $\boldsymbol{\pi}_j$, and is not used as the cluster distribution, as was done by Bouguila and Ziou [23, 24].

The intention is to use this model for general clustering applications, and so a specific distribution for the observations is not assumed here (in the experiments, Gaussian clusters, with a Gaussian-Wishart prior, were found to be most effective). So, the observations, $\mathbf{x}_{jn}$, can be drawn from any exponential family distribution given a mixture component k. Its parameters, θ_k, are drawn from a conjugate prior distribution

with hyper-parameters η and $\boldsymbol{\nu}$,

$$p(\mathbf{x}_{jn}|\theta_k) = f(\mathbf{x}_{jn})g(\theta_k)\exp\{\boldsymbol{\phi}(\theta_k)^\top \mathbf{u}(\mathbf{x}_{jn})\}, \tag{4.4}$$

$$p(\theta_k|\eta,\boldsymbol{\nu}) = h(\eta,\boldsymbol{\nu})g(\theta_k)^\eta \exp\{\boldsymbol{\phi}(\theta_k)^\top \boldsymbol{\nu}\}. \tag{4.5}$$

Here $g(\theta_k)$ and $h(\eta,\boldsymbol{\nu})$ are log-partition or normalisation functions, $\boldsymbol{\phi}(\theta_k)$ are natural parameters, $\mathbf{u}(\mathbf{x}_{jn})$ are sufficient statistics of the data, and $f(\mathbf{x}_{jn})$ is a function of $\mathbf{x}_{jn}$.

4.3 Variational Bayes for learning the Model

The goal of variational Bayes [5] is to tractably approximate the log-marginal likelihood of a model, $\log p(\mathbf{X}) = \log \int p(\mathbf{X},\mathbf{Z},\boldsymbol{\Pi},\boldsymbol{\Theta})\, d\mathbf{Z} d\boldsymbol{\Pi} d\boldsymbol{\Theta}$, for performing Bayesian inference. This approximation is known as *free energy*, and lower-bounds the log-marginal likelihood, which is optimised for a set of model hyper-parameters. The derivation of this optimisation procedure follows the standard treatment for exponential family models as presented in [9] and Section 2.2.

The derivations are started by approximating the true posterior over the parameters, $p(\mathbf{Z},\boldsymbol{\Pi},\boldsymbol{\Theta}|\mathbf{X})$, with a family of factorised mean-field approximating distributions,

$$q(\mathbf{Z},\boldsymbol{\Pi},\boldsymbol{\Theta}) = \prod_{k=1}^{K} q(\theta_k) \times \prod_{j=1}^{J} q(\boldsymbol{\pi}_j) \prod_{n}^{N_j} q(z_{jn})\,. \tag{4.6}$$

Following Beal [9], the negative free energy is,

$$\begin{aligned}\mathcal{F}[q(\mathbf{Z})\,, q(\boldsymbol{\Pi},\boldsymbol{\Theta})] = \sum_{k=1}^{K} \mathbb{E}_{q_\theta}\left[\log \frac{q(\theta_k)}{p(\theta_k|\eta,\boldsymbol{\nu})}\right] \\ + \sum_{j=1}^{J} \mathbb{E}_{q_{\boldsymbol{\pi}}}\left[\log \frac{q(\boldsymbol{\pi}_j)}{\mathrm{GDir}(\boldsymbol{\pi}_j|\mathbf{a},\mathbf{b})}\right] - \sum_{j=1}^{J}\sum_{n=1}^{N_j} \mathcal{L}_{jn}.\end{aligned} \tag{4.7}$$

It is important to note that the free energy terms involving $\boldsymbol{\pi}_j$ actually only have $K-1$ degrees of freedom. This is because the Kth term in the generalised Dirichlet

distribution is a function of the other weights, as can be seen in Equation 4.3. The term $\mathcal{L}$ is similar to an expected log-likelihood term for an observation with respect to the variational parameters,

$$\mathcal{L}_{jn} = \log \sum_{k=1}^{K} \exp \left\{ \mathbb{E}_{q_\pi}[\log p(z_{jn} = k|\pi_{jk})] + \mathbb{E}_{q_\theta}[\log p(\mathbf{x}_{jn}|\theta_k)] \right\}. \tag{4.8}$$

The expectation $\mathbb{E}_{q_\pi}[\log p(z_{jn} = k|\pi_{jk})]$ is obtained from evaluating a Categorical distribution using its expected parameters. These expectations are given in Appendix A. By minimising Equation 4.7, fitting the model hyper-parameters to the data in Equation 4.8 is regularised by full Bayesian model complexity penalty terms [9] to provide a simple model which explains the data.

For inference, the probability of an observation belonging to a cluster needs to be evaluated. An analytical expression for the variational posterior label probabilities can be derived by taking the functional derivative $\partial\mathcal{F}/\partial q(\mathbf{Z}) = 0$, while using Lagrange multipliers to enforce $\int q(\mathbf{Z}) d\mathbf{Z} = 1$. This results in the variational Bayes expectation (VBE) step,

$$q(z_{jn} = k) = \exp \left\{ \mathbb{E}_{q_\pi}[\log p(z_{jn} = k|\pi_{jk})] + \mathbb{E}_{q_\theta}[\log p(\mathbf{x}_{jn}|\theta_k)] - \mathcal{L}_{jn} \right\}, \tag{4.9}$$

here $\mathcal{L}_{jn}$ acts as a normalisation constant. Also required is a way of updating the parameters of the model conditioned on the observations. By taking functional derivatives $\partial\mathcal{F}/\partial q(\mathbf{\Pi}) = 0$ and $\partial\mathcal{F}/\partial q(\mathbf{\Theta}) = 0$, while enforcing similar normalisation constraints, the variational Bayes maximisation (VBM) step is obtained. This leads directly to the following variational posterior hyper-parameter updates,

$$\tilde{a}_{jk} = a_k + \sum_{n=1}^{N_j} q(z_{jn} = k)\,, \tag{4.10}$$

$$\tilde{b}_{jk} = b_k + \sum_{n=1}^{N_j} \sum_{l=k+1}^{K} q(z_{jn} = l)\,, \tag{4.11}$$

$$\tilde{\eta}_k = \eta + \sum_{j=1}^{J} \sum_{n=1}^{N_j} q(z_{jn} = k)\,, \tag{4.12}$$

$$\tilde{\boldsymbol{\nu}}_k = \boldsymbol{\nu} + \sum_{j=1}^{J} \sum_{n=1}^{N_j} q(z_{jn} = k)\, \mathbf{u}(\mathbf{x}_{jn})\,. \tag{4.13}$$

The variational posterior parameter distributions have the same form as the priors, i.e. $q(v_{jk}) = \text{Beta}(v_{jk}|\tilde{a}_{jk}, \tilde{b}_{jk})$, and $q(\theta_k) = p(\theta_k|\tilde{\eta}_k, \tilde{\boldsymbol{\nu}}_k)$, which has the same form as Equation 4.5. The sum in Equation 4.11 for $\tilde{b}_{jk}$ needs to be performed in descending mixture weight order in a similar fashion to [126] and [63]. The expectations present in all of these equations are given in Appendix A. To learn this model and cluster the data, the VBE and VBM steps are iteratively cycled until the negative free energy of the model Equation 4.7 converges to a local minimum.

Variational Bayes can automatically eliminate superfluous clusters, however it cannot explicitly create clusters. It is quite common to randomly initialise these types of algorithms with a large number of clusters, and then let Variational Bayes only use populate clusters it has evidence for, while the rest revert to their prior values, and can be deleted [12]. However, it has been established in the literature [9, 63], that guiding the search for clusters can actually help avoid these algorithms converging to degenerate local minima. To this end, the exhaustive cluster splitting heuristic used by Kurihara et al. [63] for cluster creation is also implemented in this work and detailed in Algorithm Algorithm 4.1. This algorithm starts with $K = 1$, and successively splits the clusters until the free energy of the model is no longer improved. In the case of Gaussian clusters, the observations belonging to a cluster with $q(z_{jn} = k) > 0.5$ 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 expected free energy of the split in Algorithm Algorithm 4.1, $\mathbb{E}[\mathcal{F}_{split,k}]$, is acquired by running variational Bayes for one iteration with the new split using all of the observations. It has been found that this cluster search heuristic nearly always outperforms random initialisation.

Algorithm 4.1: The GMC exhaustive model selection heuristic

Data: Observations $\mathbf{X}$
Result: Probabilistic assignments $q(\mathbf{Z})$ and $\{\tilde{\mathbf{a}}, \tilde{\mathbf{b}}, \tilde{\eta}, \tilde{\boldsymbol{\nu}}\}$

$\{\mathbf{a}, \mathbf{b}, \eta, \boldsymbol{\nu}\} \leftarrow$ `CreatePriors()`;
$q(\mathbf{Z}) \leftarrow \{\mathbf{1}\}_{j=1}^{J}$; `// initialises with` $K = 1$

while *true* **do**
  $q(\mathbf{Z}), \{\tilde{\mathbf{a}}, \tilde{\mathbf{b}}, \tilde{\eta}, \tilde{\boldsymbol{\nu}}\}, \mathcal{F} \leftarrow$ `VarBayes(`$\mathbf{X}, q(\mathbf{Z}), \{\mathbf{a}, \mathbf{b}, \eta, \boldsymbol{\nu}\}$`)`;
  for $k = 1$ **to** K **do**
    $\mathbf{X}_{split,k} \leftarrow \{\mathbf{x}_{jn} \in \mathbf{X} : q(z_{jn} = k) > 0.5\}$;
    $q(\mathbf{Z}_{split,k}) \leftarrow$ `ClusterSplit(`$\mathbf{X}_{split,k}$`)`;
    $q(\mathbf{Z}_{split,k}) \leftarrow$ `VarBayes(`$\mathbf{X}_{split,k}, q(\mathbf{Z}_{split,k}), \{\mathbf{a}, \mathbf{b}, \eta, \boldsymbol{\nu}\}$`)`; `// refine`
    $q(\mathbf{Z}_{aug,k}) \leftarrow$ `AugmentLabels(`$q(\mathbf{Z}), q(\mathbf{Z}_{split,k})$`)`; `// add in split labels`
    $\mathbb{E}[\mathcal{F}_{split,k}] \leftarrow$ `VarBayes(`$\mathbf{X}, q(\mathbf{Z}_{aug,k}), \{\mathbf{a}, \mathbf{b}, \eta, \boldsymbol{\nu}\}$`)`; `// 1 iteration`
  $best \leftarrow \arg\min_k \{\mathbb{E}[\mathcal{F}_{split,k}]\}_{k=1}^{K}$;
  if $(\mathcal{F} - \mathbb{E}[\mathcal{F}_{split,best}])/\mathcal{F} < C_{threshold}$ **then**
    break;
  else
    $q(\mathbf{Z}) \leftarrow q(\mathbf{Z}_{aug,best})$;

4.4 Model Variants

To quantify the value of using a generalised Dirichlet prior over the mixture weights, the GMC has also been formulated to use a symmetric Dirichlet prior, $\boldsymbol{\pi}_j \sim \mathrm{Dir}(\boldsymbol{\pi}_j|\alpha)$[1]. This variant is called the symmetric grouped mixtures clustering model (S-GMC), and when specified for Categorical observations with Dirichlet priors, it is exactly smoothed LDA. It is important to note that the posterior Dirichlet on the weights is no longer necessarily symmetric. Also the S-GMC does not require updating of any of its variational hyper-parameters in cluster-size order, unlike the GMC. The reader is referred to [5, 12] and Appendix A for the variational updates to the Dirichlet distribution.

As detailed in the following sections, it is common for some clusters to have probabilistically less than one observation in certain groups. In these cases, the learning

[1] A scalar hyper-parameter input here means the same value is used for all hyper-parameters.

algorithm runtime can be reduced by not evaluating Equation 4.9 for the absent cluster, k, and explicitly setting $q(z_{jn} = k) = 0$ for all n in the relevant group, j. The jth group's contribution for the kth variational posterior parameter updates is also left out in Equation 4.10 to Equation 4.13. These are referred to as the "*sparse*" variants of the GMC and S-GMC. This method is similar to the sparse method suggested in [80] for speeding up expectation maximisation (EM) algorithms.

4.5 Image Representation

The sparse code spatial pyramid matching (ScSPM) framework of [128] is used to create image descriptors. A codebook of 1024 elements is used, which is learned from 50,000 randomly selected, 16×16 SIFT patches. The ScSPM descriptors use overlapping image patches with a stride of 8 pixels as in [128]. The resulting descriptor length is reduced to 20 dimensions with PCA. Whitening did not appear to improve results – this may be because the pooled sparse codes over the normalised SIFT features are already all of a similar scale.

As discussed in Chapter 3, the original ScSPM is not scalable to large datasets since the sparse coding method used is quite slow. The larger photo albums and AUV datasets cannot not use this descriptor as is. Following the work of [33], and the experiments in Chapter 3, it has been established that replacing the sparse coding encoder with the faster orthogonal matching pursuit (OMP) (code from [2]) is possible. Similarly, the pre-learned Caltech-101 dictionary from [128] can be used with little to no reduction in clustering performance.

Gaussian clusters are used to model the image observations, $\mathcal{N}\left(\mathbf{x}_{jn}|\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k^{-1}\right)$. These Gaussian clusters have Gaussian-Wishart priors,

$$\boldsymbol{\Lambda}_k \sim \mathcal{W}(\boldsymbol{\Lambda}_k|\boldsymbol{\Omega}, \rho) \quad \text{and} \quad \boldsymbol{\mu}_k \sim \mathcal{N}\left(\boldsymbol{\mu}_k|\mathbf{m}, (\gamma\boldsymbol{\Lambda}_k)^{-1}\right).$$

The variational posterior hyper-parameters and expectations are similar to those presented in [5, 12], and are in Appendix A.

4.6 Experiments

In this section the GMC and its variants are compared to other clustering algorithms, which are:

Variational Dirichlet process (VDP) [63]. This is similar to a Bayesian mixture model (such as the Bayesian Gaussian mixture model (BGMM) [5, 12]), but has a Dirichlet Process prior, as opposed to a Dirichlet prior on the mixture weights. This model has been used in [50, 103] for clustering imagery, and is similar to the model shown in Figure 4.2b.

Self tuning spectral clustering (ST-SC) [130] using a sparse similarity matrix and the Eigen-gap heuristic to choose K.

Gaussian mixture model with Bayes information criterion (GMM+BIC) to select the best value of K. This is learned with the EM algorithm.

Four datasets are used for this comparison;

1. The 8-class outdoor scenes dataset from [84].
2. A subset of classes from Caltech-101 [42].
3. A large visual dataset obtained from multiple AUV surveys of a deep photic zone reef in Tasmania, Australia [125].
4. A novel dataset comprising photo albums from popular tourist destinations, mostly obtained from *Flickr*, but based on the author's holidays.

A simple generalised Dirichlet prior is chosen for the GMC by setting its hyper-parameters $\mathbf{a} = \mathbf{b} = \mathbf{1}$. For the S-GMC the hyper-parameter for the Dirichlet is set $\alpha = 1$, also the VDP concentration parameter prior is set to 1. For the clusters, semi-informative prior hyper-parameters are chosen; $\rho = D$, $\mathbf{\Omega} = (\rho\lambda^{max}_{\mathrm{cov}(\mathbf{X})}C_{width})^{-1}\mathbf{I}_D$, $\mathbf{m} = \mathrm{mean}(\mathbf{X})$, and $\gamma = 1$. Here D is the dimensionality of the data, $\lambda^{max}_{\mathrm{cov}(\mathbf{X})}$ is

the largest Eigenvalue of the covariance of the data, and C_{width} is left as a tunable parameter that encodes the a-priori "width" of the mixtures. Apart from the priors $\mathbf{m}$ and $\mathbf{\Omega}$, the values of all of the priors were chosen to be the minimum valid integer value allowed by their respective distributions. This was primarily for simplicity, and changing these values had minimal impact on the final results, especially compared to the semi-informative priors.

The clustering results were compared to ground truth (human created) labels using the *V-measure* of [92], which is the same as the more common normalised mutual information (NMI) criterion of [105]. These measures have previously been used in Chapter 3, and do not require each clustering solution to have the same number of clusters as the ground truth classes. They require no manual reconciliation step, making for a fair comparison. Furthermore, it is useful to also analyse the components of V-measure to further tease apart clustering solutions.

In the absence of ground-truth, five-fold cross validation with a held-out average log-likelihood,

$$\hat{\mathcal{L}} = \frac{1}{\sum_j N_j} \sum_{j=1}^{J} \sum_{n=1}^{N_j} \log \left(\sum_{k=1}^{K} \mathbb{E}_q[\boldsymbol{\pi}_j] \mathcal{N}(\hat{\mathbf{x}}_{jn} | \mathbb{E}_q[\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k]) \right), \tag{4.14}$$

was used to quantify performance, where $\hat{\mathbf{x}}_{jn}$ is the held out data. All parameter expectations, $\mathbb{E}_q[\cdot]$, are with respect to the variational or maximum-likelihood posteriors learned using the rest of the dataset. This is very similar to the *perplexity* measure ($\exp\{-\hat{\mathcal{L}}\}$) commonly used in the natural language processing literature [17, 109], and measures an algorithm's ability to generalise to new data. In a sense it is a kind of "internal" cluster cohesiveness, or self-similarity, metric. Higher likelihoods are better. Unfortunately this metric cannot be used with ST-SC because it is not a generative model of a similar form as the Gaussian mixtures, and is not readily applicable to new data.

The first two and last experiments were performed on a 2.8 GHz Core 2 Duo, and the AUV on a 3.0 GHz Core 2 Duo. The GMC, variants, and VDP are all implemented in multi-threaded C++ (though only a single thread is used in these experiments)

and share code, so comparing their runtime is as fair as possible.

4.6.1 Number and Composition of Albums/Groups

The aim of this experiment is to explore how the structure of the groups can improve the clustering solution. The 8-class outdoor scene dataset of [84] is used, which has 2688 colour 256×256 pixel images. This is only a relatively small dataset, and so use of the original ScSPM framework of [128] is feasible. The sparse variants of the GMC and S-GMC are not used in this experiment for clarity.

This dataset does not have a natural group structure. Nor for that matter do any other standard computer vision datasets. For this reason, this dataset had to be *artificially* divided into groups. Two novel datasets with real groups are presented later. This allows for *thorough* examination of how different groups structures affect clustering. To this end, the hypothesis to be tested is that if each group of data has different proportions of the 8 ground truth classes within it, then this may present novel views of the observations in feature space to the clustering algorithm, improving its performance. Taking this further, if there are only subsets of the 8-classes in each group, overlap between these classes may effectively be removed in feature space, making it even easier for an algorithm to find the true clustering solution.

Three artificial group-types or partitions are created in this dataset;

Proportional consists of partitions of the data that all have the same proportions as the original dataset of the 8-classes within them. It is expected that there will be no significant advantage in using this dataset over the no-groups case, as no novel views of the observations are presented.

Random has groups of data constructed from divisions of the original dataset with random proportions of the 8-classes.

Subset is similar to random, but some of the proportions are also randomly set to zero, with the remainder renormalised. This effectively excludes some of the

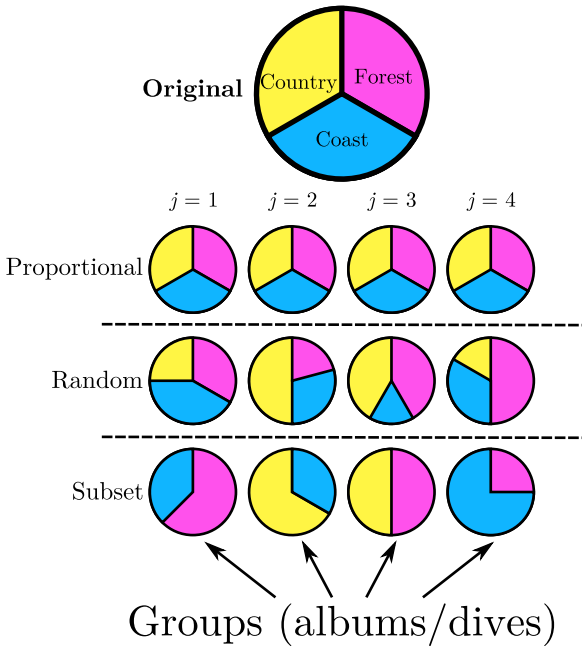


Figure 4.3 – Example of the types of artificial groups constructed; class proportions within each group (pie charts) are shown.

8-classes from some of the random groups. This is the most "natural" group type, and most closely matches the photo albums analogy out of the three group types. It is also the one in which the best performance is expected to be achieved.

An illustrative example of the structure of these groups is given in Figure 4.3. *It is important to note that the* random *and* subset *group types require hand labels to create. In the typical usage scenario for unsupervised algorithms these splitting mechanisms are not possible. This is done here so a controlled experiment investigating how group structure affects clustering can be performed.* See Section 4.6.3 and Section 4.6.4 for datasets with real group structures.

In the experiment, test sets with 1 to 20 groups for each group-type are created. Images are sequentially (not randomly) assigned to these groups. The proportional group-type test sets have no random component, and so only one test set for each constituent group number is created. The random and subset groups use 20 random trials for each group number.

Only the GMC and S-GMC can actually make use of these groups, so the original dataset is clustered using the VDP, ST-SC, and GMM+BIC algorithms for comparison. The GMC, S-GMC and VDP used a prior cluster width, C_{width}, of 0.04. Empirically this gave a good result for all algorithms. An Eigen-gap threshold of 0.025 is used for ST-SC which also empirically gave the best result. The GMM+BIC does not have a tuning parameter per-se, but it is initialised from (the best of 20 random starts of) K-means for each $K \in \{2, \ldots, 20\}$.

The clustering results of the ST-SC, GMM+BIC and VDP are summarised in Table 4.1. The corresponding results of the GMC and S-GMC for each constituent group number and for each group-type are summarised in Figure 4.4, and benchmarked against the VDP. Exemplars of the 10 classes are shown in Figure 4.5 as well as image samples from a GMC result with 10 groups of the subset type. The GMC and S-GMC have almost identical results to the VDP for one group. This is to be expected since, in this situation, the GMC is very similar to the VDP, and the S-GMC is very similar to a regular BGMM [5, 12].

To compare these algorithms with more traditional classification methods, 30 images from each class were used to train a linear support vector machine (SVM) classifier with ScSPM features as per [128]. The SVM had **84.38**% accuracy (averaged over all classes) and NMI = **0.6958** on the test images.

This experiment clearly demonstrates that leveraging grouped-data can, in principle, improve clustering results. Furthermore the hypothesis seems consistent with the results – the structure of the groups entirely influences the quality of clustering that

Table 4.1 – Summary of non-group clustering (and classification) models for the outdoor scenes dataset.

Algorithm	NMI	K
ST-SC	0.6401	12
GMM+BIC	0.6639	6
VDP	0.6854	10
GMC-subset (best)	0.7762	9.35
SVM	0.6958 (84.38%)	N/A

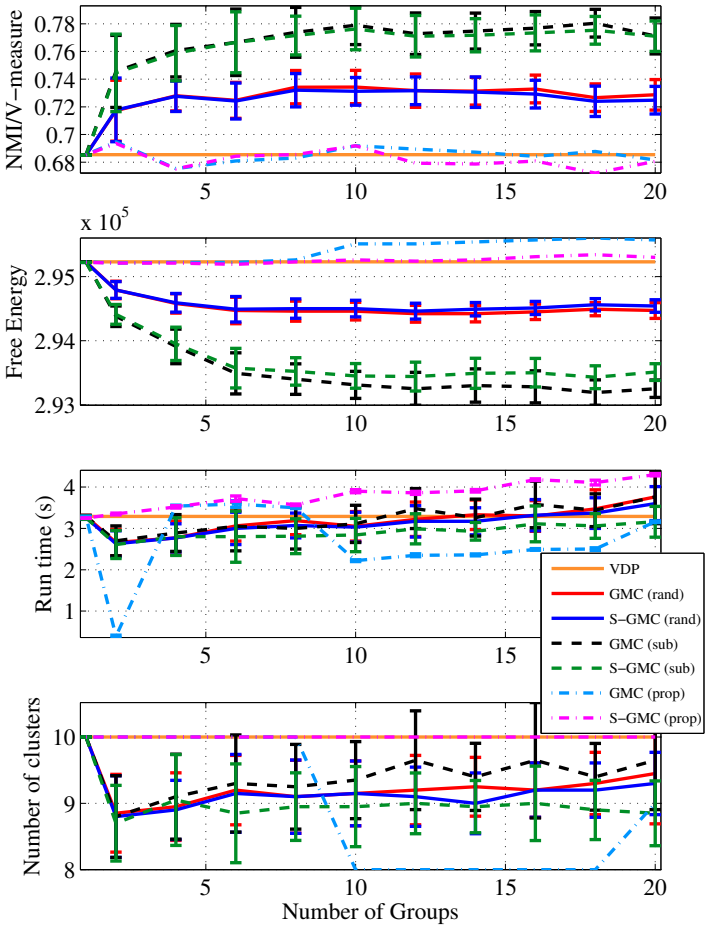


Figure 4.4 – Results of clustering the outdoor scenes dataset with an increasing number of artificially constructed groups. Lines with error bars summarise 20 randomly generated test sets. Lines without error bars have no random component. Care must be taken when comparing free energy, as it does not necessarily correspond between different clustering models, but may be used to compare the same models across group-types, where a lower value is better.

can be achieved. Interestingly, the NMI and model free energy tend to plateau after approximately 8 groups for both the random and subset group types, suggesting there may be a critical number of groups after which little more information is obtained. This experiment also suggests structured artificial dataset splitting may be useful for improving classification performance.

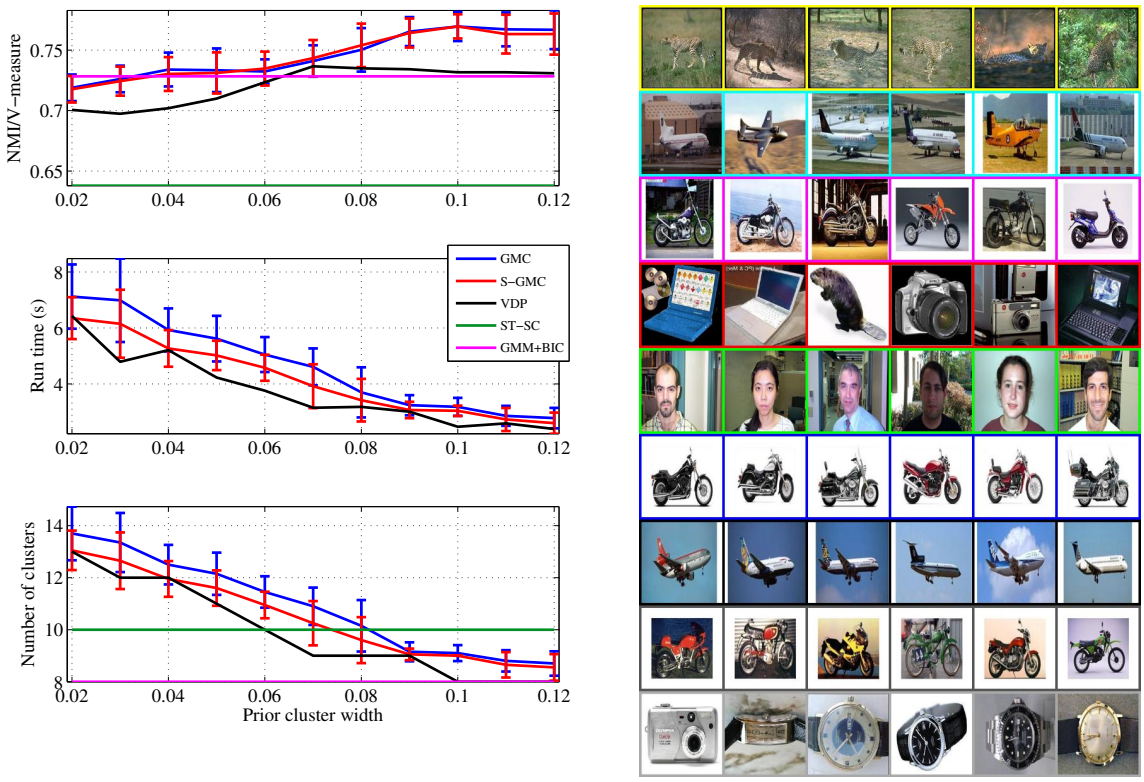
Figure 4.5 – Exemplars from the 8 classes of the outdoor scenes dataset (left), and 15 random samples (row-wise) from the 9 GMC clusters (right) with 10 groups (subset type), achieving a NMI of 0.7741.

4.6.2 Effect of Cluster Hyper Parameter Values

In this experiment, the influence of the cluster width tuning parameter, C_{width} (defined in Section 4.6), has on the clustering results is explored, since it is the only significant "knob" in these Bayesian algorithms.

For this experiment 10 classes from the 101 class Caltech-101 dataset [42] were used. Only 10 classes were used because it is unreasonable to expect a clustering algorithm to find much in common with the original labelled data for so many classes, especially when many have relatively few examples. This has also been done commonly in the literature [50, 117], and so makes this study more comparable. The classes used are (with the number of images); Cougar-body (47), Leopards (200), Laptop (81), Camera (50), Faces (435), Airplanes (800), Motorbikes (798), Watch (239), Elephant (64), Beaver (46). This dataset is not very large, and so it was feasible to use the same ScSPM feature encoder as the last experiment.

The *subset* group type, with 10 groups, is used again here for the GMC and S-GMC since it gave the best results in the last experiment. However, the proportions of



(a) (b)

Figure 4.6 – (a) Results of clustering 10 of the Caltech 101 classes in 10 artificial groups. The prior cluster width tuning parameter is varied along the x-axis. Lines with error bars summarise 20 test sets. (b) shows 6 random samples (row-wise) from the 9 GMC clusters with $C_{width} = 0.1$, achieving a NMI of 0.7756. The confusion between the smaller classes is most likely because these classes do not have enough evidence to be assigned their own clusters for this setting of C_{width}.

the ground truth classes in the groups had to be fixed to reflect those of the original dataset. So only ground truth classes were randomly removed from each of the 10 groups. This had to be done because the original classes varied so much in size that quite often no random solution could be found for the original subset group type.

The C_{width} parameter is varied from 0.02 to 0.12. Again 20 random trials are performed for each increment to use with the GMC and S-GMC. The sparse variants are not included for clarity. It was found the best Eigen-gap threshold for ST-SC was 0.01, and a search for $K \in \{2, \ldots, 20\}$ was performed for the GMM+BIC. The results are summarised in Figure 4.6a and sample clusters from one of the better

GMC results are presented in Figure 4.6b.

The GMC and S-GMC achieve roughly the same NMI in this experiment, and consistently outperform the VDP, and ST-SC. The GMM+BIC algorithm initially outperforms all other algorithms when C_{width} is small, however it is quickly surpassed by the GMC and S-GMC. The runtime for the VDP is lower than the other algorithms, however this is a relatively small dataset, and the overhead of managing the groups may be overwhelming any time advantage from the GMC and S-GMC. Again this is compared to a linear SVM classifier and ScSPM features with 30 training images, which obtained a **91.36**% accuracy and NMI = **0.9197** on the remaining images. Interestingly, the SVM does not appear to suffer from the inconsistent error problem which may have plagued the last experiment.

Like in the last experiment with the subset groups, the GMC and S-GMC perform only slightly differently. That is, the GMC finds more clusters, and consequently has a longer runtime than the S-GMC, despite the similar NMI. The differences are likely because of the small datasets used in these experiments, which allow for the priors to exert more influence. This effect of the prior is almost completely overwhelmed by the data-likelihood in the following experiments, which have more data.

4.6.3 Case Study on a Scientific Dataset

For this experiment a dataset obtained from stereo cameras on an AUV was used. The dataset has approximately 100,000 stereo image pairs from 12 survey dives (used as the groups) over rocky reefs near the Tasman National Park on the East coast of Tasmania, Australia [124, 125]. The images are of the various habitats on the seafloor and are taken at a target altitude of 2 metres. The modified ScSPM framework was used to encode the monochrome images of the pair. The images were reduced from 1360×1024 to 320×241 pixels, and it took about one second per image to extract descriptors.

This dataset had 9 image classes; sand/reef interface, low relief reef, coarse sand, patch reef, fine sand, screw shell rubble ($> 50\%$), screw shell rubble ($< 50\%$), high

relief reef, and Ecklonia (Kelp). Every single image in this dataset had an associated label, provided by marine scientists [100]. Unfortunately this dataset was found to have a very large proportion of incorrect labels. 6000 images were chosen at random from all of these dives for correction according to a provided labelling key. All images were clustered, but only these 6000 were used for validation.

An Eigen-gap threshold of 0.0025 gave the best results for ST-SC, and $K \in \{2, \ldots, 30\}$ was searched for the GMM+BIC. The results are summarised in Figure 4.7 for varying prior cluster widths. Presented in Figure 4.8 are sample images from the best S-GMC result with exemplars from hand labelled classes. The GMC cluster weights for each dive, $\boldsymbol{\pi}_j$, are also overlaid on a map of the region. Separately, $\hat{\mathcal{L}}$ was calculated for the VDP and GMC, shown in Figure 4.9.

The GMC and S-GMC consistently converge in *substantially less time* than the VDP, and achieve either similar or substantially better NMIs. The sparse variants of these algorithms also converge in a little less time than the full versions, without compromising NMI in most cases. It is interesting to see that with a lot of data, the choice of prior for the GMC and S-GMC, has little effect on the clustering result. In a similar fashion, the variational BGMM algorithm [12] was also run on this dataset, but the results are not shown because they were indistinguishable from the VDP. For large datasets the model likelihood terms, especially those of the Gaussian clusters, increasingly dominate the effects of the group weight priors in the free energy objective function. This leads to similar clustering solutions for algorithms that use Gaussian clusters[2]. From a Bayesian standpoint, the large dataset is probably the reason why the VDP and GMC variants perform more similarly than in the previous experiments for NMI; there is far more evidence for clusters. Though the GMC mostly creates more cohesive and generalisable clusters than the VDP according to $\hat{\mathcal{L}}$. And the runtime difference still accords with the hypothesis that taking advantage of the natural group structure makes inference easier.

All algorithms have fairly low NMIs. This may be because the 9 labelled classes

[2]This phenomenon may be because of the Gaussian observation model. LDA and HDP typically are used with a Categorical observation models (words) [17, 109], which may be more affected by the choice of mixture weight priors. More on this in the next chapter.

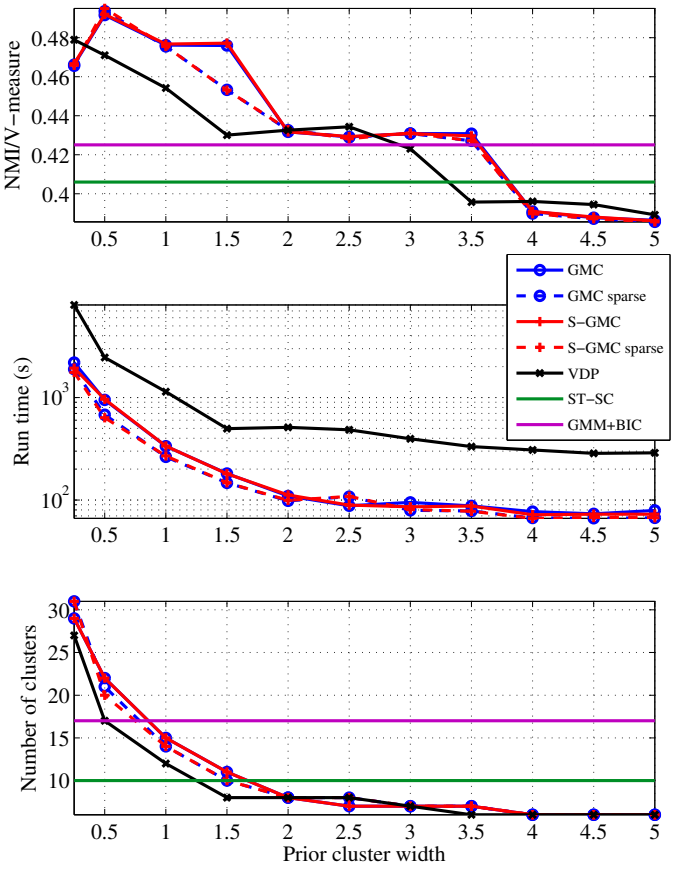
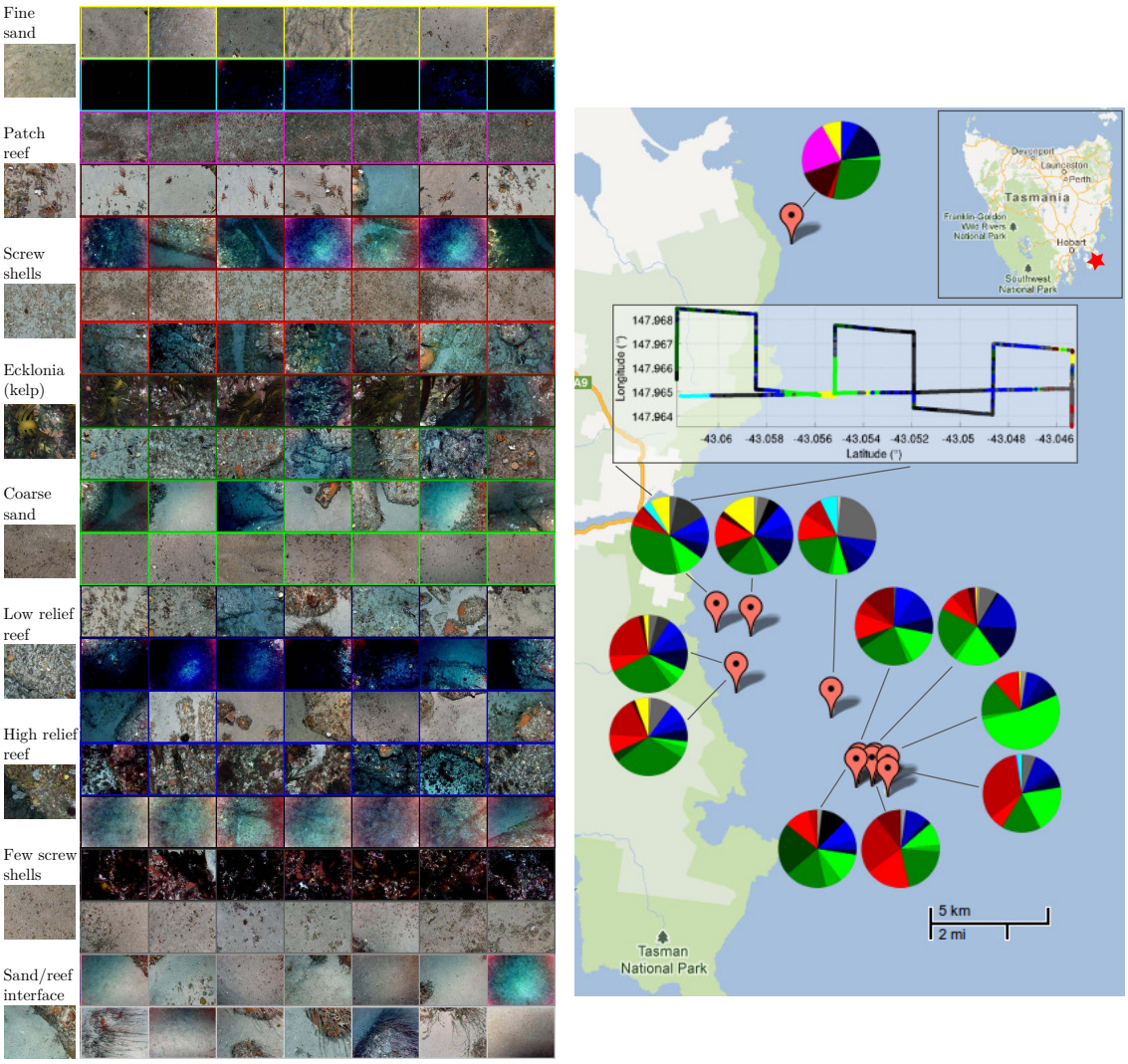


Figure 4.7 – Prior cluster width, C_{width}, effects on NMI, run time (log scale) and number of clusters.

do not sufficiently capture the wide variety of bottom types encountered, and are consequently quite inhomogeneous. Furthermore, some of the class differences are not immediately apparent from the visual data. For example, the difference between patch and low relief relies on neighbouring images to determine the extent of the reef. If it is "small" it is a patch reef. Also, in some images illumination is poor enough that textural information is lost, which affects the SIFT descriptors in the ScSPM, and impacts results. These factors lead to a low cluster-class correspondence. Despite the low NMI, the clusters generated by the GMC and S-GMC look visually consistent. Furthermore, their ability to summarise each survey by its cluster weights,



(a) Truth exemplars and cluster samples. **(b)** Group weights, $\boldsymbol{\pi}_j$, for all dives.

Figure 4.8 – AUV survey experiment examples. (a) shows exemplars of the 9 ground truth classes, and samples from the 20 S-GMC (0.5) clusters, NMI = 0.495. The images in these samples are randomly chosen from *all* of the surveys. In (b) the S-GMC cluster weights (pie charts) for each dive (markers) are overlaid on a map of the region – courtesy of Google Maps, location is indicated by the red star. Also shown is the vehicle path for dive 1. Each dot is the location of a stereo pair coloured by cluster label. The colours in (b) correspond to the coloured frames in (a). It can be seen that using the GMC to cluster this whole campaign leads to quite a compact summary of the imagery.

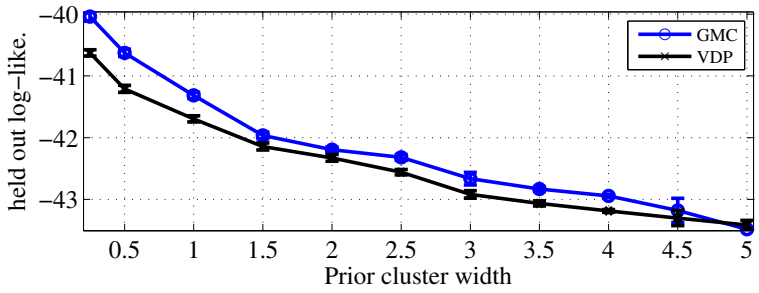


Figure 4.9 – C_{width} effects on held-out log-likelihood, $\hat{\mathcal{L}}$, for the AUV dataset.

as in Figure 4.8, provides a valuable starting point for marine scientists in ascribing semantic content to the images, and focusing on particular subsets of interest for further analysis.

4.6.4 Case Study on a Photo Collection

For the final experiment a dataset of 12 photo albums was constructed based on holidays of the author. Approximately 2200 photos are from the author, and 8100 images from *Flickr* (creative commons), downloaded from the same locations based on relevance. This dataset was used as is, with only panoramas, and personal/sensitive photos removed. The albums are; Barcelona (690), Blue Mountains (810), Bodrum & Ephesus (1046), Boston (1209), Dublin & Kilkenny (929), Istanbul (655), Marlborough Sound (778), Milford Sound (1180), New Hampshire (512), Research Cruises (545), San Francisco and Los Angeles (1129), and Taipei (842). None of these images had class-specific ground-truth labels, so NMI cannot be used. The images were scaled to have a maximum dimension of 320 pixels (aspect preserved), and the modified ScSPM descriptors were used.

Results are presented in Figure 4.10. This dataset is less constrained in terms of the diversity of scene types, and their inherent proportions, than the other datasets. Because of this, and the lack of ground-truth, the most likely and least likely samples from each cluster with tags from Flickr are shown in Figure 4.11 for the GMC and Figure 4.12 for the VDP. This is to clarify where these algorithms are succeeding and failing. Random samples do not portray this as clearly for this dataset (shown

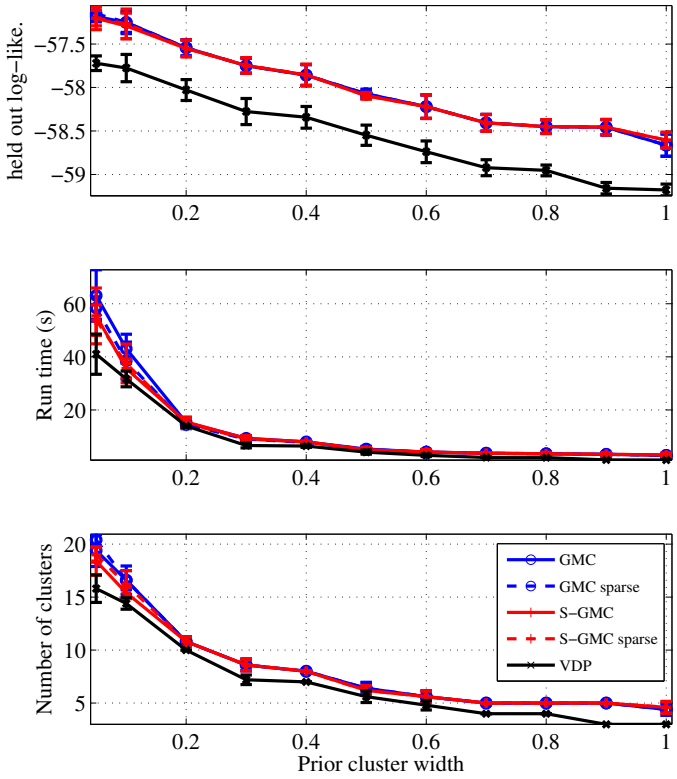


Figure 4.10 – Results for the holiday albums dataset. The GMM+BIC results are $\hat{\mathcal{L}} = -63.55$ (1.16), and an average $K = 8.8$ (0.45). They have not been plotted for clarity.

in Figure 4.13 and Figure 4.14). It is also expected that in many cases this is how end-users will view the output of these algorithms.

The GMM+BIC has a $\hat{\mathcal{L}} = \mathbf{-63.55}$ (**1.16**), and an average $K = \mathbf{8.8}$ (**0.45**), this is worse than the other models tested. According to this measure, the GMC variants have more cohesive and generalisable clusters than the VDP and GMM+BIC, which is also reflected in the visual samples. This is not a very large dataset, and the GMC variants always find more clusters, which is costly on run-time, so they are marginally slower than the VDP. The sparse GMC variants provide no apparent advantage here.

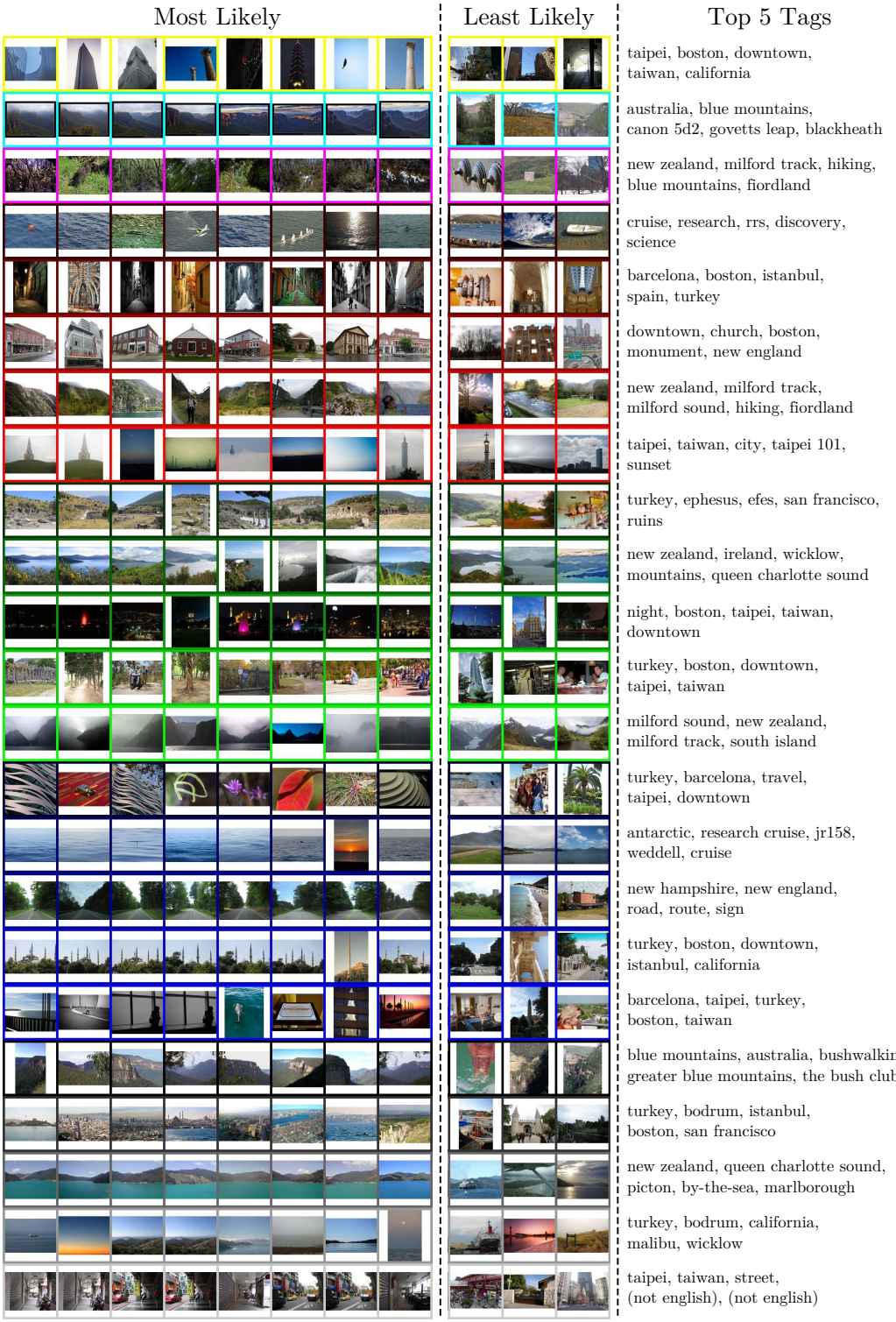


Figure 4.11 – Examples of the clusters (row-wise) from the GMC on the holiday albums dataset, $C_{width} = 0.05$. The top 5 tags by occurrence in all images in each cluster are also shown. Images are ranked by probability, $p(z_{jn} = k)$.

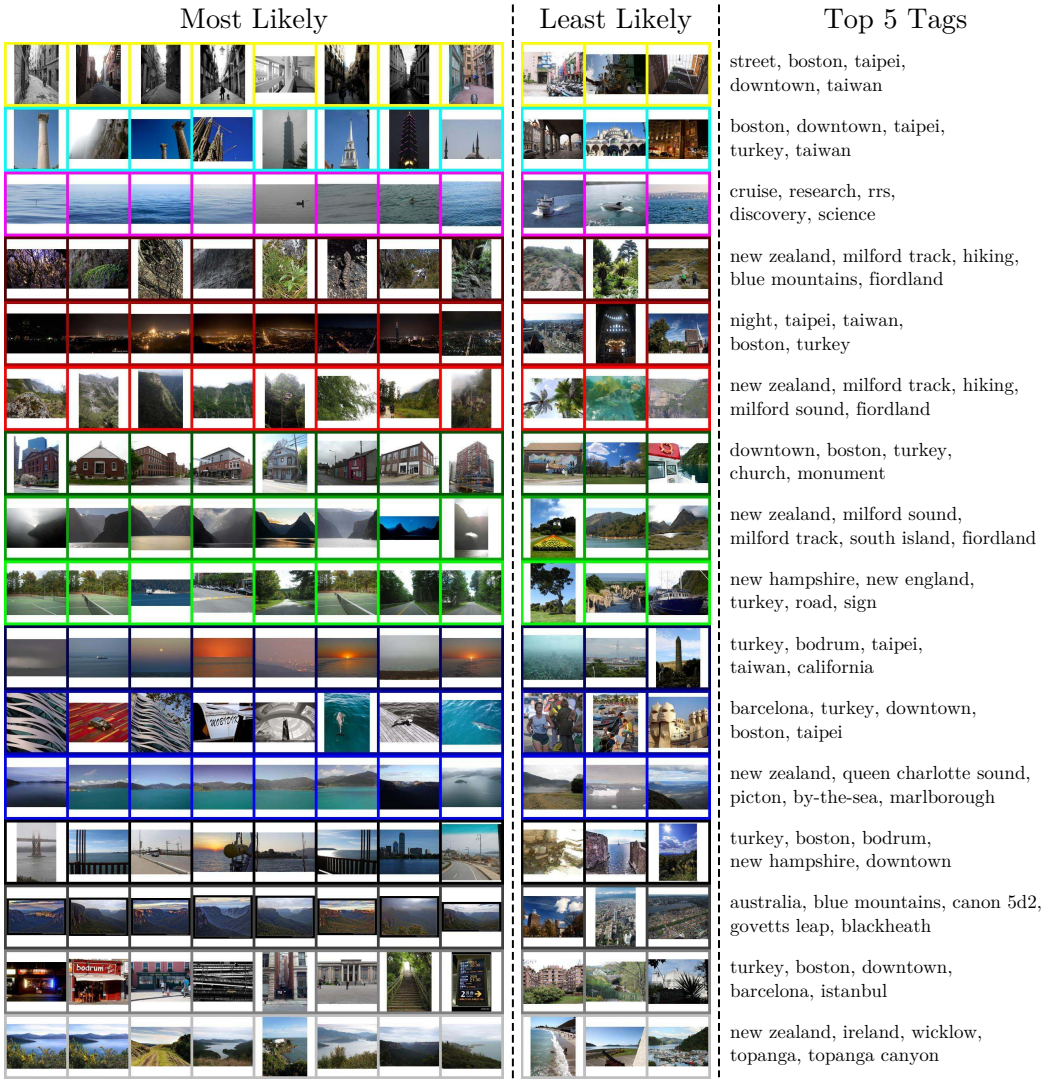


Figure 4.12 – Examples of the clusters (row-wise) from the VDP on the holiday albums dataset, $C_{width} = 0.05$. The top 5 tags by occurrence in all images in each cluster are also shown. Images are ranked by probability, $p(z_n = k)$.

4.7 Summary

There appears to be little to no difference in clustering performance between the GMC and S-GMC variants, as quantified by NMI. However, in the smaller datasets, the generalised Dirichlet prior seemed to find more clusters. This difference vanished almost entirely with the larger datasets, suggesting that as the models accrued evidence, the effects of the group weight priors were overwhelmed by the strong, and dominantly

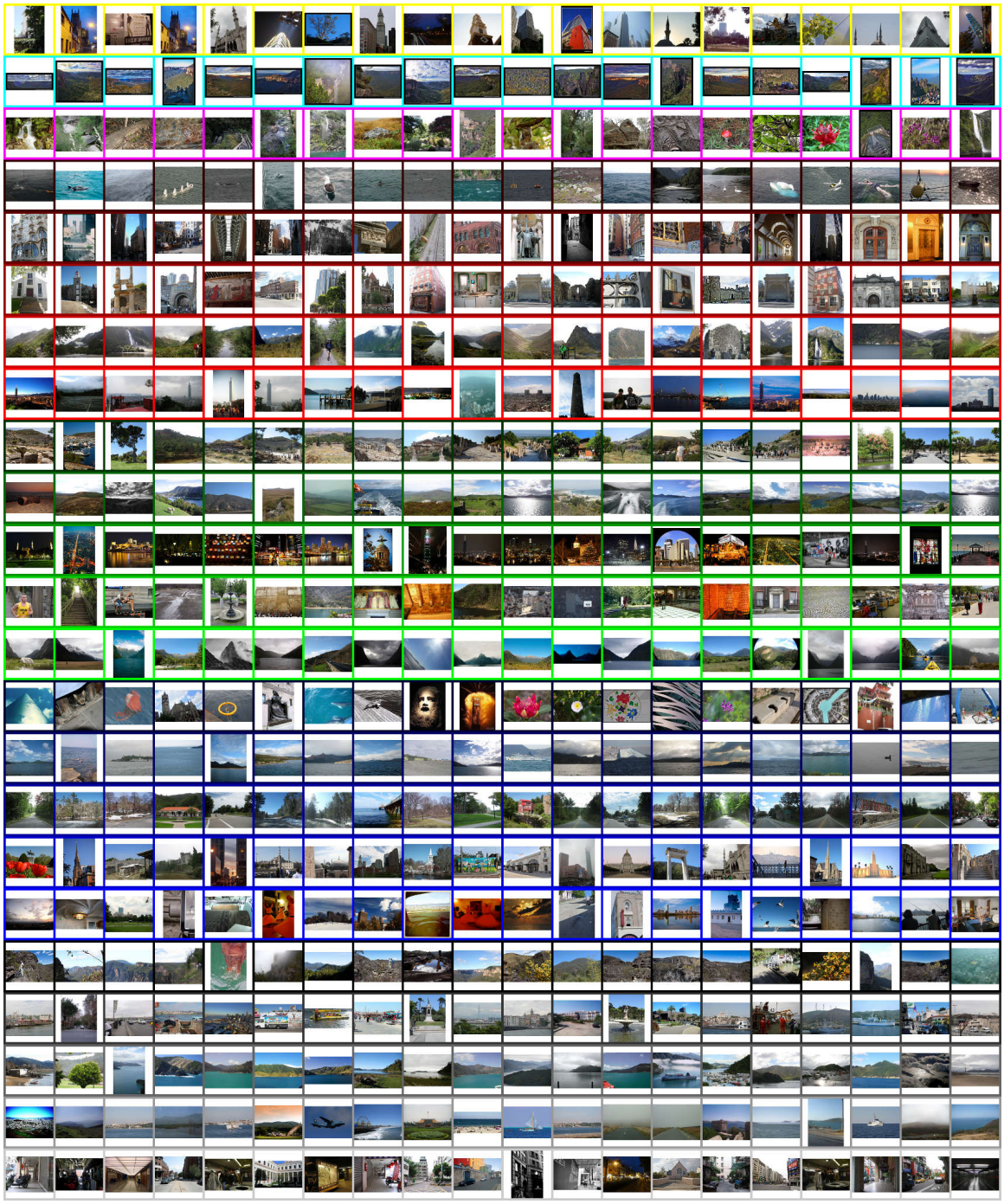


Figure 4.13 – Examples of the clusters (row-wise) from the GMC on the holiday albums dataset, $C_{width} = 0.05$. These images are randomly selected.

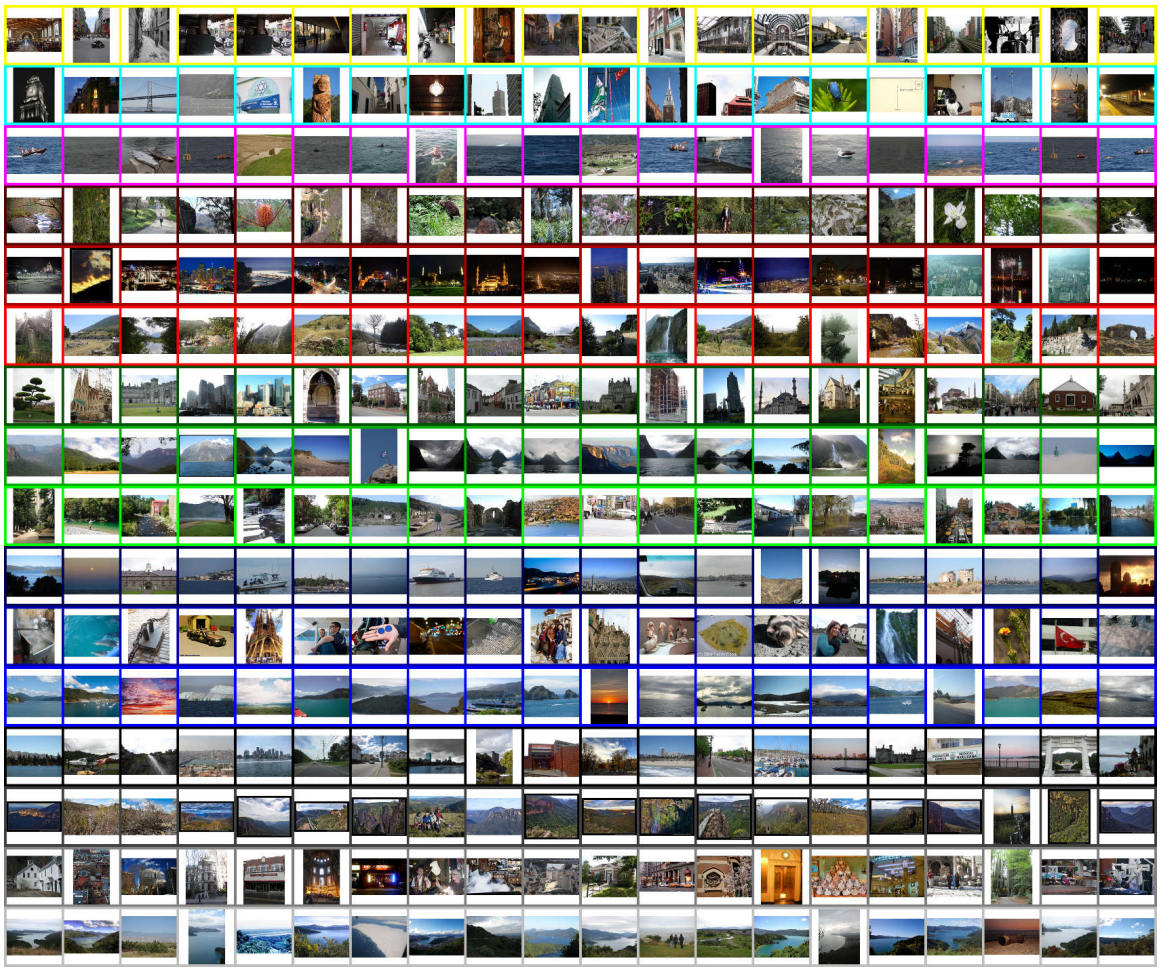


Figure 4.14 – Examples of the clusters (row-wise) from the VDP on the holiday albums dataset, $C_{width} = 0.05$. These images are randomly selected.

Gaussian, likelihood terms. Thus it is hard to recommend using one choice of prior over the other. If simplicity is desired, then the symmetric Dirichlet may be more appropriate. If it is known a-priori that the cluster distribution is asymmetric, and the dataset it small, then a generalised Dirichlet may be more appropriate.

In the first experiment it was observed both the NMI and negative free energy plateauing with respect to the number of groups. This suggests there may be a critical number of groups after which there is no information gain. As future work it would be interesting to try establish what properties inherent in the groups of data create this plateauing effect.

From these experiments it is quite clear that by modelling the distributions of im-

ages in their groups or albums, which share clusters, the GMC variants outperform normal clustering models. It has also been shown that the more distinct the proportions of clusters within the groups are from one another, the better these algorithms perform. For large datasets there is a significant reduction in runtime compared to more conventional Bayesian mixture models. The reason for these improvements is that groups of related data provide more view points of the observations in feature space. These additional view-points expose separation between clusters of data that are not apparent when no distinction is made between groups. This is particularly evident when clusters may overlap in feature space, but do not co-occur in the same group. Essentially, the GMC can model and take advantage of the context in which the observations occur to provide better clustering solutions.

As future work it would be interesting to replace single-membership image clusters with some kind of multi-membership or factor model. That is, each image can belong to multiple clusters, or factors (with positive mixing only). The analogy being that natural scenes may be composed of factors such as “mountain”, “lake” etc., where no one factor may best describe the scene. In the underwater imagery, this may be observed as images being composed of factors such as “sand” and “reef” etc. One model that may be capable of achieving this would be a multi-level HDP, if sub-image features where clustered. A model similar to this is considered in the next chapter.

Chapter 5

Clustering Multiple Levels of Related Visual Datasets

In this chapter the grouped mixtures clustering model (GMC) from Chapter 4 is extended in order to further explore how modelling context effects clustering. The new model now simultaneously clusters images and segments, or super-pixels, within images, while also jointly clustering over groups, or albums. Image clusters are defined by the proportions of segment clusters within their constituent images. These image clusters essentially model simple "object" co-occurrence, and provide context for segment clusters. Groups provide context for both image and segment clusters. These different notions of context essentially give multiple views of observations in feature space, as was discovered in the previous chapter. The number of image and segment clusters is discovered through variational Bayesian model selection. This model is compared to the GMC and a Bayesian Gaussian mixture model (BGMM) for clustering segments. It is found that this model is the fastest to cluster the datasets tested. It also provides segment clustering solutions that are competitive with the GMC, and better than the BGMM, which does not model context at all. The contribution of this chapter is in providing a thorough empirical understanding of how the structure of the models presented and choice of prior distributions affect clustering in a *fully unsupervised* Bayesian setting.

5.1 Introduction

The aim of this chapter is to explore and compare various ways of modelling "contextual" information in images, using *unsupervised* Bayesian modelling techniques. In the previous chapter it was found that including the notion of "albums" or groups when clustering multiple visual datasets, was advantageous. This was because these groups gave novel views of observations in feature space, which simplified the task of clustering. In this chapter, the structure of these models is extended to now simultaneously cluster image segments *while* clustering images. It is shown that image clusters can also provide a contextual benefit to clustering segments as groups did to images in the previous chapter. Furthermore, the notion of an album or group context is retained at the image cluster level. A diagram representing what is being modelled, and what is meant by simultaneous clustering in groups, is shown in Figure 5.1.

Similar models to the one proposed here have been used before in unsupervised and supervised vision tasks [39, 69, 106, 108], and in [127] for EEG seizure modelling. Most of these models use Bayesian non-parametric priors, and all are far more complex in structure than the models presented here, but are also more capable. For instance, Du et al. [39] used a spatial non-parametric process to take into account the spatial layout of segments within an image. Li et al. [69] present a model that simultaneously learns a dictionary and encodes image features, though it does not explicitly segment the images. Both models can use annotated data, where available, in a semi-supervised manner. However both of these models require substantial computational effort to learn compared to those presented here. They also do not include a notion of separate groups or albums, and no thorough analysis is done on the choices of model structure, and priors used, in fully unsupervised settings. There has been prior work in unsupervised object discovery by Russell et al. [95], Tuytelaars et al. [117]. This is similar in spirit to this work, but this work instead focuses on scene recognition via holistic image modelling. [95] is focused primarily on unsupervised object discovery/segmentation and image retrieval. Their method of combining multiple segmentations with topic models yields some very visually similar objects. In [117] many clustering and latent-variable methods are compared for unsupervised

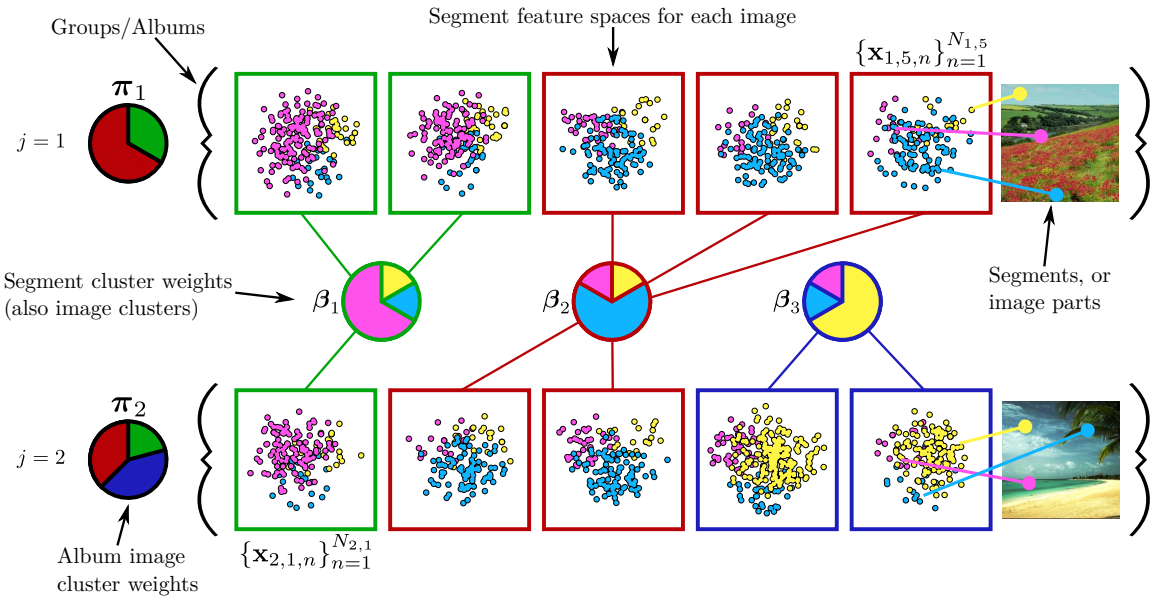


Figure 5.1 – Demonstration of simultaneous clustering in groups. Each point is the segment or super-pixel observation, $\mathbf{x}_{jin}$, in feature space. Each square is a feature space corresponding to one image, indexed by i. Images occur in albums, or groups (curly braces), indexed by j. Segments are clustered into "objects" (cyan $\approx$ plant, magenta $\approx$ water and yellow $\approx$ sky), and are shared between images and groups. Images with a similar proportion and co-occurrence of segment clusters, $\boldsymbol{\beta}_t$, form the image clusters (red, green and blue squares). Groups can be described by the proportions of image clusters within them, $\boldsymbol{\pi}_j$. As in Chapter 4 there are multiple views of the observations in feature space, which simplifies inference.

object discovery. They concluded that unsupervised object discovery, where there are multiple objects per scene, is a difficult and largely unsolved problem. The work presented in this chapter provides further insights into this problem.

Apart from [39, 106] there has been much work on using spatial context within images to improve segmentation and object recognition [4, 107, 122, 132]. Similarly, Torralba et al. [115] uses temporal based smoothing from a hidden Markov model (HMM) to improve the classification of indoor scenes from a video stream. Interestingly, Torralba et al. [115] and Sudderth et al. [106] also use the image class, as given by an "oracle", to increase the *a priori* probability of certain objects appearing within an image. Choi et al. [31] take this further, and use spatial location, and object co-occurrence hierarchies as context to improve object detection. The models considered in this chapter do not explicitly use spatial or temporal smoothing (apart from the

over-segmented image regions used as input), but instead try to obtain good segmentation from the use of strong image features, and context from relatively unexplored *unsupervised* means (image clusters, albums etc.). Spatial smoothing or context is not modelled primarily since there is usually a large computational cost associated with it.

Many other semi-supervised and supervised probabilistic models have been created to leverage context present within images from human annotations/tags as well as other contextual sources, [30, 70–72]. This is a very active research area since it is common to tag photos on sites such as *Flickr*, which ideally could provide an excellent resource of essentially free training data. In some applications, such as the underwater dataset obtained from an autonomous underwater vehicle (AUV) presented here, these annotations are not easily obtained, since they may require expert knowledge to generate. In these cases it is important to consider fully unsupervised models. There is little in the literature exploring fully unsupervised, annotation-less models for simultaneous clustering, and so this work attempts to fill this gap.

The models presented here use simple parametric priors on the mixture weight distributions, such as Dirichlet and generalised Dirichlet [36] distributions, as opposed to non-parametric priors such as the Dirichlet Process [43]. Both Dirichlet and generalised Dirichlet priors have been used to approximate Dirichlet Processes in variational Bayes inference [15, 123] and, depending on the model structure, can achieve fairly similar results. For this reason, and to avoid the need for invoking a complex hierarchical Dirichlet process (HDP) prior over groups [109], parametric priors have been chosen for the models presented in this chapter. A thorough empirical analysis of the effects of the choice of generalised versus symmetric Dirichlet prior is presented.

In the next section, the generative model for simultaneous clustering in groups is presented. In Section 5.3 a variational Bayes learning algorithm is derived, and a greedy model selection heuristic presented. In Section 5.4 some variants of the model are discussed, and in Section 5.5 the features used to describe images are presented. Experiments on standard datasets, a scientific dataset obtained from an AUV, and the photo albums dataset from Chapter 4 are presented in Section 5.6. The results

are summarised in Section 5.7.

5.2 Clustering Multiple Levels of Images Over Multiple Datasets

In this section the simultaneous segment and image clustering model is introduced, and will be referred to as the simultaneous clustering model (SCM). Rather than specifying a separate weight distribution of observation clusters, or "topics" for each "document" or image, as in latent Dirichlet allocation (LDA) [17], it specifies one for a cluster of like images. It infers these clusters of images by finding images that exhibit similar proportions and co-occurrences of segment clusters within them. Hence, the SCM uses an image-cluster as the context for searching for segment ("word") clusters, as opposed to LDA-like models that use an individual image as the context. Furthermore, the SCM has a notion of group or "album" context, which is implemented in a similar fashion as the LDA image context, but at a higher level like the GMC in Chapter 4. This is summarised in Figure 5.1.

Observations, or image segments $\mathbf{x}_{jin} \in \mathbb{R}^D$, are assumed to be arranged in the following manner:

- There are N_{ji} segments in each image, $\mathbf{X}_{ji} = \{\mathbf{x}_{jin}\}_{n=1}^{N_{ji}}$.
- There are I_j images in a group, or "album", $\mathbf{X}_j = \{\mathbf{X}_{ji}\}_{i=1}^{I_j}$.
- There are J groups, or albums, in the whole dataset, $\mathbf{X} = \{\mathbf{X}_j\}_{j=1}^{J}$.

The aim is to discover K segment clusters, parametrised by $\boldsymbol{\Theta} = \{\theta_k\}_{k=1}^{K}$, shared between all of the images, and T image clusters, parametrised by $\mathbf{B} = \{\boldsymbol{\beta}_t\}_{t=1}^{T}$, shared between the groups. The t-th image cluster parameters are just proportions of segment clusters, $\boldsymbol{\beta}_t = [\beta_{t1}, \ldots, \beta_{tk}, \ldots, \beta_{tK}]$, where $\beta_{tk} \in [0, 1]$ and $\sum_k \beta_{tk} = 1$. Furthermore, the j-th group or album is described by the proportions of the image clusters within it, $\boldsymbol{\pi}_j = [\pi_{j1}, \ldots, \pi_{jt}, \ldots, \pi_{jT}]$, again $\pi_{jt} \in [0, 1]$ and $\sum_t \pi_{jt} = 1$.

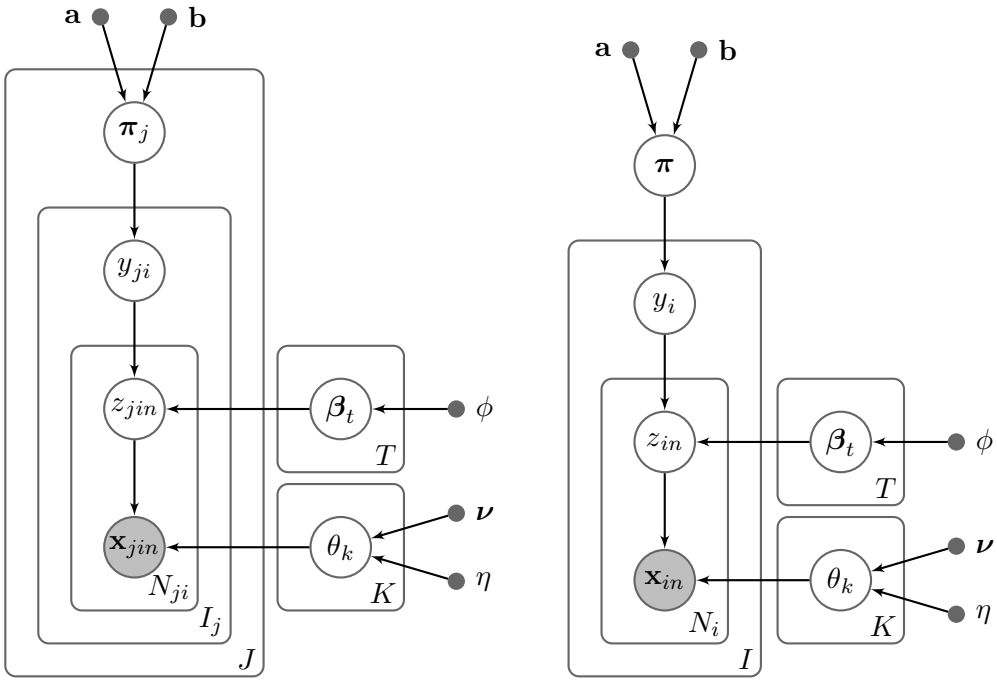
Latent auxiliary variables are used for assigning images to image clusters, y_{ji}, and segments to segment clusters, z_{jin}. Once the cluster parameters have been drawn; $\theta_k \sim p(\eta, \boldsymbol{\nu})\,\forall k$ and $\boldsymbol{\beta}_t \sim \mathrm{Dir}(\phi)\,\forall t$ [1], the following generative process for this model is assumed for a group, j,

1. Draw group mixture weights, $\boldsymbol{\pi}_j \sim \mathrm{GDir}(\mathbf{a}, \mathbf{b})$.
2. For each of the I_j images in group j,
   (a) Choose an image cluster, $y_{ji} \sim \mathrm{Cat}(\boldsymbol{\pi}_j)$, where $y_{ji} \in \{1, \ldots, T\}$.
   (b) For each of the N_{ji} segments in image ji,
      i. Choose a segment cluster, $z_{jin} \sim p\big(y_{ji}, \mathbf{B}\big)$, from a Categorical distribution with parameters $\mathbf{B}$ indexed by y_{ji}, where $z_{jin} \in \{1, \ldots, K\}$.
      ii. Draw an observation, $\mathbf{x}_{jin} \sim p(z_{jin}, \boldsymbol{\Theta})$, from an exponential family distribution with parameters $\boldsymbol{\Theta}$ indexed by the label, z_{jin}.

The collection of all of the group mixture weights is termed $\boldsymbol{\Pi} = \{\boldsymbol{\pi}_j\}_{j=1}^{J}$. This generative process is somewhat reminiscent of the author-topic model of Steyvers et al. [104], but the SCM would have one "author" for each "document", using the analogies common in the text modelling literature. The graphical model for this generative process is given in Figure 5.2a, and the corresponding joint distribution is,

$$p(\mathbf{X}, \mathbf{Y}, \mathbf{Z}, \boldsymbol{\Pi}, \mathbf{B}, \boldsymbol{\Theta} | \mathbf{a}, \mathbf{b}, \phi, \eta, \boldsymbol{\nu}) = \prod_{k=1}^{K} p(\theta_k | \eta, \boldsymbol{\nu}) \prod_{t=1}^{T} \mathrm{Dir}(\boldsymbol{\beta}_t | \phi) \times \prod_{j=1}^{J} \mathrm{GDir}(\boldsymbol{\pi}_j | \mathbf{a}, \mathbf{b}) \prod_{i=1}^{Ij} \mathrm{Cat}\big(y_{ji} | \boldsymbol{\pi}_j\big) \prod_{n=1}^{N_{ji}} p\big(z_{jin} | y_{ji}, \mathbf{B}\big)\, p(\mathbf{x}_{jin} | z_{jin}, \boldsymbol{\Theta})\,. \quad (5.1)$$

[1] A scalar hyper-parameter argument in $\mathrm{Dir}(\cdot)$ or $\mathrm{GDir}(\cdot, \cdot)$ means the same value is used for all hyper-parameters.



(a) SCM – Groups **(b)** SCM – No groups

Figure 5.2 – The SCM for multiple datasets (a), and for one dataset (b) – see Section 5.4.

The last two terms can be further factorised,

$$p\big(z_{jin}|y_{ji},\mathbf{B}\big) = \prod_{t=1}^{T} \mathrm{Cat}(z_{jin}|\boldsymbol{\beta}_t)^{\mathbf{1}[y_{ji}=t]} \tag{5.2}$$

$$p(\mathbf{x}_{jin}|z_{jin},\boldsymbol{\Theta}) = \prod_{k=1}^{K} p(\mathbf{x}_{jin}|\theta_k)^{\mathbf{1}[z_{jin}=k]} \tag{5.3}$$

Recall from Chapter 4 that $\mathbf{1}[\cdot]$ is an indicator function that returns 1 when the condition in the brackets is true, and 0 otherwise. A Generalised Dirichlet distribution [36, 126] is used over the group mixture weights, $\mathrm{GDir}(\boldsymbol{\pi}_j|\mathbf{a},\mathbf{b})$. This results in fewer image clusters as opposed to a Dirichlet distribution prior – more detail on this choice is in Section 5.6.3. It is essentially the same as a truncated stick-breaking

process [13, 59],

$$\pi_{jt} = v_{jt}\prod_{s=1}^{t-1}(1-v_{js}), \qquad v_{jt} \sim \begin{cases} \text{Beta}(a_t, b_t) & \text{if } t < T \\ 1 & \text{if } t = T, \end{cases} \tag{5.4}$$

where $v_{jt} \in [0, 1]$ are 'stick-lengths' for each group. For generality and clarity, the observations, $\mathbf{x}_{jin}$, are assumed to be drawn from any exponential family distribution given a segment mixture component k (again, Gaussian clusters are used in the experiments). Its parameters, θ_k, are drawn from a conjugate prior distribution with hyper parameters η and $\boldsymbol{\nu}$,

$$p(\mathbf{x}_{jin}|\theta_k) = f(\mathbf{x}_{jin})g(\theta_k)\exp\{\boldsymbol{\phi}(\theta_k)^\top \mathbf{u}(\mathbf{x}_{jin})\}, \tag{5.5}$$

$$p(\theta_k|\eta, \boldsymbol{\nu}) = h(\eta, \boldsymbol{\nu})g(\theta_k)^\eta \exp\{\boldsymbol{\phi}(\theta_k)^\top \boldsymbol{\nu}\}. \tag{5.6}$$

Here $g(\theta_k)$ and $h(\eta, \boldsymbol{\nu})$ are log-partition or normalisation functions, $\boldsymbol{\phi}(\theta_k)$ are natural parameters, $\mathbf{u}(\mathbf{x}_{jin})$ are sufficient statistics of the data, and $f(\mathbf{x}_{jin})$ is a function of $\mathbf{x}_{jin}$.

5.3 Variational Bayes for Learning the Models

The derivations are started by approximating the true posterior over the parameters, $p(\mathbf{Y}, \mathbf{Z}, \boldsymbol{\Pi}, \mathbf{B}, \boldsymbol{\Theta}|\mathbf{X})$, with a family of factorised mean-field approximating distributions,

$$q(\mathbf{Y}, \mathbf{Z}, \boldsymbol{\Pi}, \mathbf{B}, \boldsymbol{\Theta}) = \prod_{k=1}^{K} q(\theta_k) \times \prod_{t=1}^{T} q(\boldsymbol{\beta}_t) \times \prod_{j=1}^{J} q(\boldsymbol{\pi}_j) \prod_{i=1}^{I_j} q\big(y_{ji}\big) \prod_{n=1}^{N_{ji}} q(z_{jin})\,. \tag{5.7}$$

Following [9], the negative free energy lower bound is,

$$\mathcal{F}[q(\mathbf{Y}, \mathbf{Z})\,, q(\boldsymbol{\Pi}, \mathbf{B}, \boldsymbol{\Theta})] = \sum_{k=1}^{K} \mathbb{E}_{q_\theta}\left[\log \frac{q(\theta_k)}{p(\theta_k|\eta, \boldsymbol{\nu})}\right]$$

$$+\sum_{t=1}^{T}\mathbb{E}_{q_{\boldsymbol{\beta}}}\left[\log\frac{q(\boldsymbol{\beta}_t)}{\mathrm{Dir}(\boldsymbol{\beta}_t|\phi)}\right]+\sum_{j=1}^{J}\mathbb{E}_{q_{\boldsymbol{\pi}}}\left[\log\frac{q(\boldsymbol{\pi}_j)}{\mathrm{GDir}(\boldsymbol{\pi}_j|\mathbf{a},\mathbf{b})}\right]$$
$$+\sum_{j=1}^{J}\sum_{i=1}^{Ij}\sum_{n=1}^{N_{ji}}\mathbb{E}_{q}\left[\log\frac{q\big(y_{ji}\big)\,q(z_{jin})}{\mathrm{Cat}\big(y_{ji}|\boldsymbol{\pi}_j\big)\,p\big(z_{jin}|y_{ji},\mathbf{B}\big)\,p(\mathbf{x}_{jin}|z_{jin},\boldsymbol{\Theta})}\right], \quad (5.8)$$

where the last term's expectation is with respect to all of the latent variables and parameters. This last term acts like a data-fitting objective, and the first three terms act as model complexity penalties. The last term does not simplify down to a "log-likelihood" like term as it does with the BGMM or GMC. This is because of the interaction between the latent variables, $\mathbf{Y}$ and $\mathbf{Z}$. The only consequence of this is that it makes it slightly harder to formulate a suitable greedy cluster splitting heuristic, as detailed later. It is also important to note that the $\mathrm{GDir}(\cdot)$ term only has $T-1$ degrees of freedom, as can be seen from Equation 5.4. All of the expectations are given in Appendix A. The learning objective is to minimise this negative free energy.

For inference, we need first to evaluate the probability of an observation belonging to a segment cluster. An analytical expression for the variational posterior label probabilities can be derived by taking the functional derivative $\partial\mathcal{F}/\partial q(\mathbf{Z})=0$, while using Lagrange multipliers to enforce $\int q(\mathbf{Z})d\mathbf{Z}=1$. This results in the variational Bayes expectation (VBE) step for the segment labels,

$$q(z_{jin}=k)=\frac{1}{\mathcal{Z}_{z_{jin}}}\exp\left\{\sum_{t=1}^{T}q\big(y_{ij}=t\big)\,\mathbb{E}_{q_\beta}[\log\beta_{tk}]+\mathbb{E}_{q_\theta}[\log p(\mathbf{x}_{jin}|\theta_k)]\right\}. \quad (5.9)$$

This is like the expectation step in a regular mixture model, but the mixing weights are themselves weighted according to the probability of the current image containing the segment belongs to each image cluster. It was feared this double mixing may cause slow convergence. However, when combined with a strong segment cluster likelihood distribution, like a Gaussian, this is not an issue.

As before, taking $\partial\mathcal{F}/\partial q(\mathbf{Y})=0$, while enforcing $\int q(\mathbf{Y})d\mathbf{Y}=1$, results in the VBE

step for the image labels – or the probability an image belongs to an image cluster,

$$q\big(y_{ji}=t\big)=\frac{1}{\mathcal{Z}_{y_{ji}}}\exp\left\{\mathbb{E}_{q_\pi}[\log\pi_{jt}]+\sum_{k=1}^{K}\mathbb{E}_{q_\beta}[\log\beta_{tk}]\sum_{n=1}^{N_{ji}}q(z_{jin}=k)\right\}. \tag{5.10}$$

This is similar to a mixture of Multinomial distributions, since each image is represented as a sum of Categorical distributions. The $\mathcal{Z}$ terms are normalisation constants, which are,

$$\mathcal{Z}_{z_{jin}}=\sum_{k=1}^{K}\exp\left\{\sum_{t=1}^{T}q\big(y_{ij}=t\big)\,\mathbb{E}_{q_\beta}[\log\beta_{tk}]+\mathbb{E}_{q_\theta}[\log p(\mathbf{x}_{jin}|\theta_k)]\right\}, \tag{5.11}$$

$$\mathcal{Z}_{y_{ji}}=\sum_{t=1}^{T}\exp\left\{\mathbb{E}_{q_\pi}[\log\pi_{jt}]+\sum_{k=1}^{K}\mathbb{E}_{q_\beta}[\log\beta_{tk}]\sum_{n=1}^{N_{ji}}q(z_{jin}=k)\right\}. \tag{5.12}$$

Again, all of the expectations used in the preceding equations are given in Appendix A.

The variational Bayes maximisation (VBM) steps are derived in the same way as the VBE steps by setting $\partial\mathcal{F}/\partial q(\boldsymbol{\Theta})=0$, while enforcing $\int q(\boldsymbol{\Theta})d\boldsymbol{\Theta}=1$, for all parameters, $\{\mathbf{B},\boldsymbol{\Pi},\boldsymbol{\Theta}\}$. Solving this leads directly to the following variational posterior hyperparameter updates,

$$\tilde{a}_{jt}=a_t+\sum_{i=1}^{I_j}q\big(y_{ji}=t\big)\,, \tag{5.13}$$

$$\tilde{b}_{jt}=b_t+\sum_{i=1}^{I_j}\sum_{s=t+1}^{T}q\big(y_{ji}=s\big)\,, \tag{5.14}$$

$$\tilde{\phi}_{tk}=\phi+\sum_{j=1}^{J}\sum_{i=1}^{I_j}q\big(y_{ji}=t\big)\sum_{n=1}^{N_{ji}}q(z_{jin}=k)\,, \tag{5.15}$$

$$\tilde{\eta}_{k}=\eta+\sum_{j=1}^{J}\sum_{i=1}^{I_j}\sum_{n=1}^{N_{ji}}q(z_{jin}=k)\,, \tag{5.16}$$

$$\tilde{\boldsymbol{\nu}}_{k}=\boldsymbol{\nu}+\sum_{j=1}^{J}\sum_{i=1}^{I_j}\sum_{n=1}^{N_{ji}}q(z_{jin}=k)\,\mathbf{u}(\mathbf{x}_{jin})\,. \tag{5.17}$$

The variational posterior parameter distributions have the same form as the prior,

i.e. $q(v_{jt}) = \text{Beta}(v_{jt}|\tilde{a}_{jt}, \tilde{b}_{jt})$, $q(\boldsymbol{\beta}_t) = \text{Dir}\left(\boldsymbol{\beta}_t|\tilde{\phi}_{t1}, \ldots, \tilde{\phi}_{tK}\right)$, and $q(\theta_k) = p(\theta_k|\tilde{\eta}_k, \tilde{\boldsymbol{\nu}}_k)$, which has the same form as Equation 5.6. The sum in Equation 5.14 for $\tilde{b}_{jt}$ has to be performed in descending mixture weight order in a similar fashion to [126] and [63]. To learn this model and cluster the data, the VBE and VBM steps are iterated until the negative free energy of the model in Equation 5.8 converges to a local minimum.

As stated in Chapter 4 variational Bayes can automatically eliminate superfluous clusters, but it cannot explicitly find new clusters. The exhaustive splitting heuristic in Chapter 4 could again be used here, which successively tries to split every cluster, and chooses the split that lowers the model free energy the most. For a large number of segment clusters this can take a very long time, so instead a greedy splitting heuristic is created that attempts to guess the best split first. The learning algorithm starts with $K = 1$, and successively splits the segment clusters using the greedy splitting heuristic until the free energy of the model is no longer improved.

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,

$$\hat{\mathcal{F}}_k = \mathbb{E}_{q_\theta}\left[\log \frac{q(\theta_k)}{p(\theta_k|\eta, \boldsymbol{\nu})}\right] - \sum_{j=1}^{J}\sum_{i=1}^{I_j}\sum_{n=1}^{N_{ji}} q(z_{jin} = k)\, \mathcal{L}_{z_{jin}=k} \tag{5.18}$$

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 the exact form of

this heuristic is,

$$\hat{\mathcal{F}}_k = \mathbb{E}_{q_\theta}\left[\log \frac{q(\theta_k)}{p(\theta_k|\eta,\boldsymbol{\nu})}\right] - \sum_{j=1}^{J}\sum_{i=1}^{I_j}\sum_{n=1}^{N_{ji}} q(z_{jin}=k)\,\mathbb{E}_{q_\theta}[\log p(\mathbf{x}_{jin}|\theta_k)]\,. \tag{5.19}$$

A cluster weight term was not included in Equation 5.19 because a corresponding term of opposite sign existed in the last term in Equation 5.8, and adding it would nullify its effect in the overall model free energy. This is not the case for the GMC presented in Chapter 4, for which this heuristic would be (with images, i, being the groups in this context),

$$\hat{\mathcal{F}}_k = \mathbb{E}_{q_\theta}\left[\log \frac{q(\theta_k)}{p(\theta_k|\eta,\boldsymbol{\nu})}\right] + \sum_{i=1}^{I}\sum_{n=1}^{N_i} q(z_{in}=k)\left[\mathbb{E}_{q_\pi}[\log \pi_{ik}] + \mathbb{E}_{q_\theta}[\log p(\mathbf{x}_{in}|\theta_k)]\right]. \tag{5.20}$$

This will be used in Section 5.6 for both the GMC and the BGMM (which does not factor over i).

How a segment cluster is split depends on its distribution. In the case of Gaussian segment clusters, the observations belonging to a cluster 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 Algorithm 5.1. 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 the author's 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, which would have required substantially better computational hardware than was used to perform the AUV experiments in Section 5.6. It also managed to maintain good clustering results compared to the exhaustive heuristic.

While this splitting heuristic may work for the segment clusters if they are Gaussian,

this is not straight forward for the image clusters, which are essentially Multinomial. Hence, the model is randomly initialised with some large number of image clusters, $T_{trunc} \gg T$, which are naturally pruned. Unfortunately this removes the deterministic nature of the algorithm, and may leave it more susceptible to converging to local extrema in the free energy functional for one particular run.

5.4 Model Variants

Figure 5.2b presents a simplified SCM that does not take into account the context provided by albums or groups. Alternatively, it can be seen as the SCM for a single group ($J = 1$). In Section 5.6, two other variants of the SCM are tested; (1) the Generalised Dirichlet prior over group mixtures is replaced with a Dirichlet prior, $\boldsymbol{\pi}_j \sim \text{Dir}(a)$, and (2) a Generalised Dirichlet prior is placed over both the group mixtures, and the image cluster parameters, $\boldsymbol{\beta}_t \sim \text{GDir}(\phi, \delta)$. The effects of these model choices are further explored in Section 5.6.

As in Chapter 4, it is common for image and segment clusters to have probabilistically less than one observation in groups and images respectively. This natural sparsity could be utilised to speed variational learning in the previous section as was done in Chapter 4. However, these sparse variants have not been implemented with the SCM, and are not tested here.

5.5 Image Representation

The sparse code spatial pyramid matching (ScSPM) image representation studied in Chapter 3 and used in Chapter 4 is not appropriate for use in modelling image-parts (or segments). It has been designed to represent the structural layout of whole images, and its patch based representation is too coarse for use here. Furthermore, it does not model local colour and fine texture variations, which may be important for modelling local image parts.

Out of the many image representations tried, it was found that pooling dense independent component analysis (ICA) [58] codes within image 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 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 the fast SLIC super pixel method [1][2].
6. Obtain segment descriptors by mean pooling all of the ICA dictionary responses in a segment in the following manner:

$$\tilde{\mathbf{x}}_{jin} = \frac{1}{\#S_{jin}} \sum_{l \in S_{jin}} \log |\mathbf{a}_l| \tag{5.21}$$

7. Obtain the final segment descriptors, $\mathbf{x}_{jin}$, by PCA whitening all the $\tilde{\mathbf{x}}_{jin}$. Usually dimensionality reduction is performed here too.

This process is graphically demonstrated in Figure 5.3

Other feature learning techniques were tried instead of ICA, such as various sparse coding techniques [2, 121, 128], with max-pooling. These sparse coding techniques learn over-complete dictionaries, which require iterative solvers to encode patches as

[2]Subsequent to the writing of this thesis, mean-shift segments [32] have been used and offer improved results to those reported here.

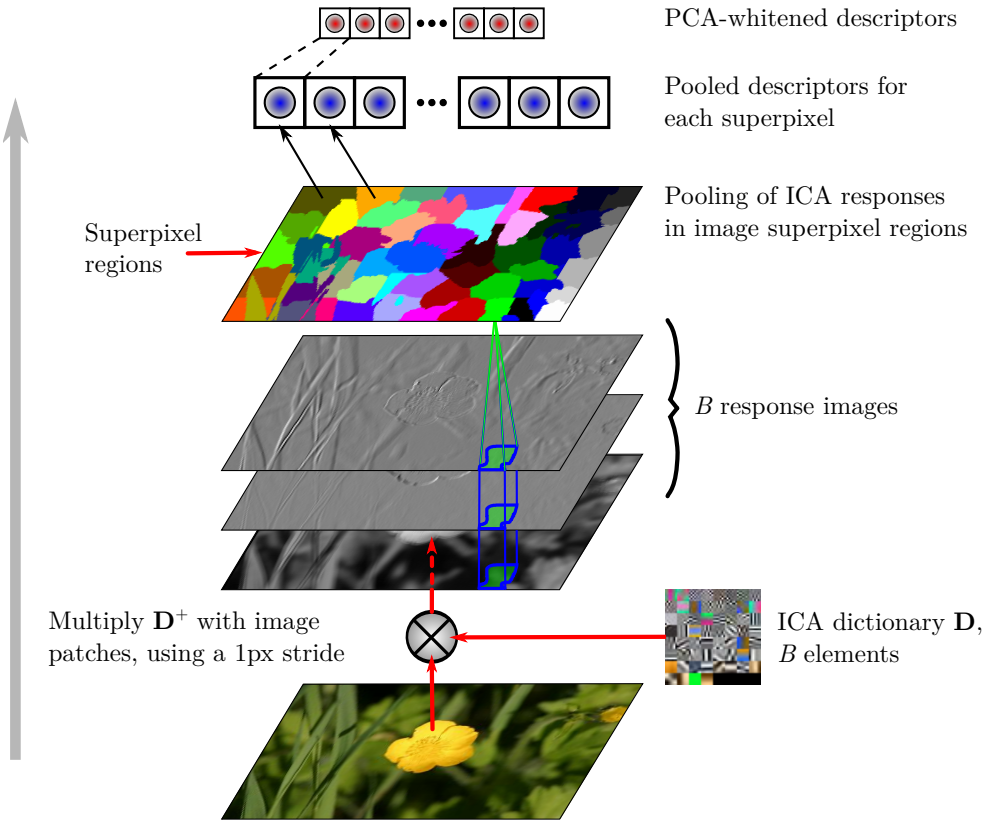


Figure 5.3 – ICA based super-pixel descriptors.

opposed to ICA's (under)-complete dictionary, which has an analytical solution. This made encoding every single pixel with sparse coding infeasible for some of the larger datasets used. Furthermore, encoding every pixel with ICA resulted in better features than using a more sophisticated sparse coding technique on a subset of pixels.

It was found that Equation 5.21 was more effective than regular mean pooling within segments. This may be because ICA produces filters that are ambiguous in terms of sign (or 90 degree phase shifts). Similarly, this pooling method is invariant to 90 degree phase shifted signals (i.e. the response to a white bar on a black background is the same as a black bar on a white background). The logarithmic transform made the absolute valued responses more normally distributed. This helped with taking the mean, and also with PCA whitening (which both have Gaussian data assumptions), and was found to have a large impact on performance. Once the dictionary had been learnt, it took approximately one second per image to extract these features (most of

the images used were approximately 300×300 pixels).

Gaussian clusters are used for the whitened segment observations, $\mathcal{N}\left(\mathbf{x}_{jn}|\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k^{-1}\right)$. These Gaussians have Gaussian-Wishart priors,

$$\boldsymbol{\Lambda}_k \sim \mathcal{W}(\boldsymbol{\Lambda}_k|\boldsymbol{\Omega}, \rho) \quad \text{and} \quad \boldsymbol{\mu}_k \sim \mathcal{N}\left(\boldsymbol{\mu}_k|\mathbf{m}, (\gamma\boldsymbol{\Lambda}_k)^{-1}\right).$$

The variational posterior hyper-parameters and expectations are similar to those presented in [5, 12], and are in Appendix A.

5.6 Experiments

In this section the SCM variants and two other similar models are compared. The other models are;

GMC from Chapter 4. This models each image as its own mixture model, but shares clusters between images. So there is a notion of "within image" context. The GMC is essentially used in the same way as LDA is used, however Categorical distributions are not used as the observation model. The greedy cluster splitting heuristic is used for this model.

BGMM [5, 12]. This is just one mixture model for all of the segments, and so has no real notion of context[3]. Again the greedy cluster splitting heuristic has been developed and used for this model.

It would be desirable to compare these models to other supervised models in the literature, such as [39, 72], in similar fashion as Chapter 3 and 4. Unfortunately there are two main obstacles to this. The first is that many of the segmentation results are presented purely qualitatively or only one object per scene. Secondly, many models are sufficiently unique to a specific dataset, or type of data, to make completely fair

[3]The variational Dirichlet process (VDP) used for comparison in Chapter 4 is not used here because there is little difference to the BGMM in terms of clustering results when many observations are clustered.

comparisons with unsupervised methods non-trivial (or the data is not in a readily accessible form).

Four datasets will be used for the comparisons; (1) the same subset of the Microsoft Research Classes v2 (MSRC-2) used in [39, 69]. (2) The outdoor scenes dataset [84], with segment labels from LabelME [96]. (3) A scientific dataset comprising images taken from multiple AUV surveys of deep photic zone reefs off of the East coast of Tasmania. (4) the photo albums dataset from Chapter 4. The first two datasets are used to illustrate the effect of modelling different levels of context has on clustering solutions, the effect of the number of groups or albums has on clustering, as well as the effects of the choice of prior distributions. The AUV and photo albums datasets are used to demonstrate these algorithms operating on larger datasets, which are partitioned into natural groups. Unfortunately, there is not a closed form solution for SCM log-likelihood, so the photo-albums experiment is purely qualitative.

Simple hyper-parameter values were used for all distributions on weights and image clusters i.e. GDir($\mathbf{1}, \mathbf{1}$) and Dir(1) for all of the models (including the BGMM and GMC). The SCM hyper-parameter ϕ can control the number of image clusters found, and so in Section 5.6.4 this parameter is varied. For the segment clusters semi-informative prior hyper-parameters were chosen; $\rho = D$, $\mathbf{\Omega} = (\rho C_{width})^{-1}\mathbf{I}_D$, $\mathbf{m} = \text{mean}(\mathbf{X})$, and $\gamma = 1$. Here C_{width} is left as a tunable parameter that encodes the a-priori 'width' of the segment clusters.

In all of the experiments, the images were segmented into approximately 50 segments of roughly similar area using [1]. The segment descriptors calculated from Equation 5.21 were PCA whitened and reduced to $D = 15$. This preserved more than 90% of the spectral power in all cases, and drastically improved cluster results.

Normalised mutual information (NMI) [105] is again used to quantify clustering results. Also, the two components of V-measure [92] (homogeneity and completeness) are explicitly used in this chapter to compare and contrast the clustering results.

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 order to

assign a segment a ground-truth label, the mode of the pixels in the segment had to be of that label type.

For all experiments, the SCM variants were run from 10 random initialisations of the image cluster indicator parameters, $\mathbf{Y}$, with an image cluster truncation level of $T_{trunc} = 100$. The GMC and BGMM are both entirely deterministic, with run-times that barely varied, so they were each run only once for experimental evaluation.

All of the probabilistic models tested are implemented in multi-threaded C++ code, and share as much code as possible. The manner in which all of the algorithms are parallelised is different, so in the interest of fairness, only one thread is used in these experiments for runtime comparison. The MSRC-2 and Outdoor Scenes datasets were run on a 2.8 GHz Intel Core 2 Duo processor, and the AUV dataset on a 3.0 GHz Core 2 Duo.

5.6.1 Contextual Effects on Image Clustering

In this section, the SCM variants, BGMM and GMC are all compared in terms of their ability to cluster image segments, and the two SCM variants are compared in terms of their ability to cluster images. Both the MSRC-2 and Outdoor Scenes are used for this purpose.

Ten image classes were used from the MSRC-2 dataset (trees, buildings, cows, faces, cars, sheep, flowers, signs, books, and chairs) with 302 images in total. Each class was comprised of approximately 30 images. There are also 15 segment classes, and the "void" class is not included. Each image in this dataset is no wider than 320 pixels, and 5×5 pixel patches gave best results. Slightly better results were obtained without DC removal or contrast normalisation of the patches. Approximately 50,000 random patches were used to train an ICA dictionary with 50 filters.

The Outdoor Scenes dataset has eight image classes (coast, forest, highway, inside city, mountain, open country, street, tall building). Forty images from each class were used, resulting in 320 images in total. This was mainly done to facilitate the number

of experiments that could be carried out. There were hundreds of segment classes in this dataset, however many of the classes were synonymous, resulting from LabelMe not restricting label descriptions. These segment classes were manually combined into 24 classes using LabelMe's Matlab toolbox. Each image in this dataset is 256 pixels in height and width, and it was found again that neither DC removal nor contrast normalisation helped results. 7×7 pixel patches were optimal, and again 50,000 patches were used to train an ICA dictionary with 60 filters.

Neither of these datasets have natural albums or groups. So following Chapter 4, the datasets were randomly split into 5 groups, with only a random subset of the true image classes in each group. This random subset division gave best results in Chapter 4. The only model that can use these groups is the original SCM model, hence this splitting was not performed for the other models.

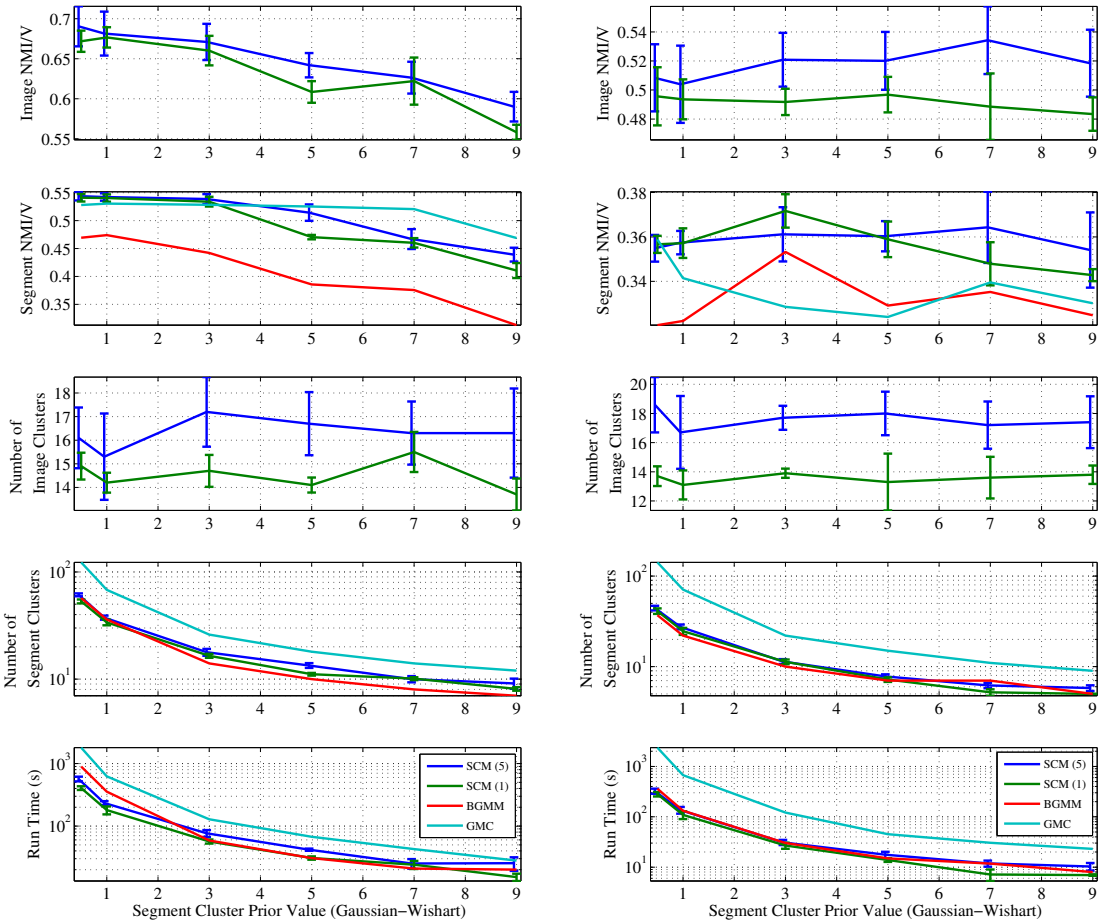
To re-iterate, the context that is being modelled by each algorithm is summarised;

1. The SCM (1), from $J = 1$, models context from image-clusters, including segment cluster co-occurrence, when clustering segments (i.e., an image cluster mainly of forest images should have a higher proportion and probability associated with leaf-like segment clusters).

2. The SCM (5), from $J = 5$, models the same context as SCM (1), but additionally models the context present in images belonging to separate albums or groups.

3. The GMC from Chapter 4 is used as an LDA style model, where every image is treated as a separate mixture model with shared segment clusters. That is, this models *each image* as having separate context when clustering segments. Since there is no notion of an image cluster, segment cluster co-occurrence is not modelled in a generalisable fashion.

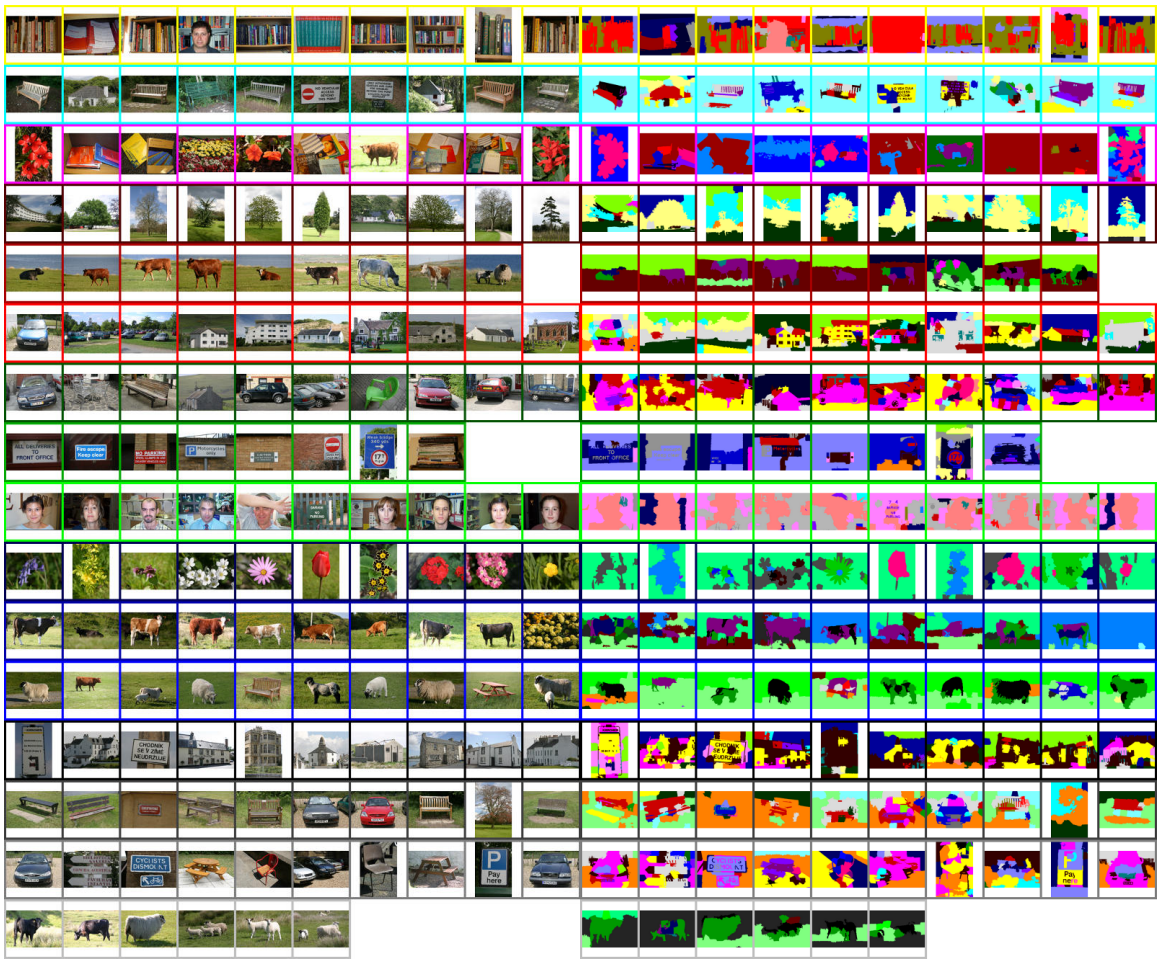
4. The BGMM has no notion of the context of images or albums, and simply clusters segments as if they were in one "bag".



(a) MSRC-2 Dataset **(b)** Outdoor Scenes Datset

Figure 5.4 – Results of modelling different context for clustering images and image segments. SCM (5) uses 5 artificial groups, SCM (1) uses only one group (the original dataset). The GMC and BGMM cannot cluster images, and the BGMM has no notion of separate images.

To thoroughly explore the performance of these models, the datasets were clustered for various values of the prior segment-cluster width tuning parameter, C_{width}. Performance of all of the models is depicted in Figure 5.4 for both datasets, and a random sample of a SCM clustering result is shown in Figure 5.5 and Figure 5.6. We can see that both SCM models perform fairly similarly for both image and segment clustering



(a) Sample image clusters **(b)** Corresponding segment clusters

Figure 5.5 – A sample SCM (1) result on the MSRC-2 dataset, with $C_{width} = 1$, and $\text{NMI}_i = 0.669$, $\text{NMI}_s = 0.551$, $K = 40$, and $T = 16$. Random image cluster samples are shown in (a) across the rows, and the corresponding segment clusters are shown in (b).

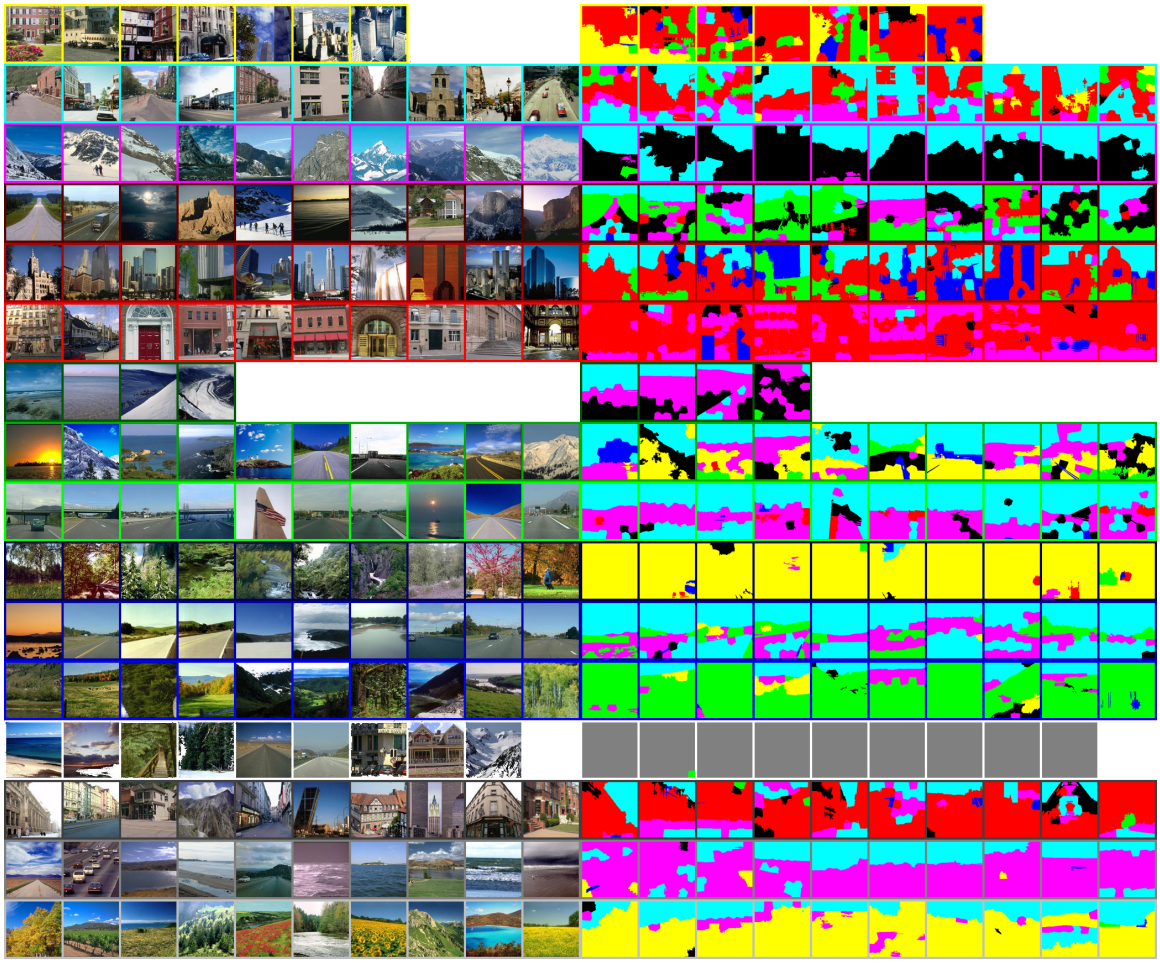
quality, with perhaps SCM (5) having a slight edge on image clustering performance. However, it also finds more image clusters fairly consistently. The effects of the number of groups on clustering is investigated more thoroughly in the next section.

We can also see that although the GMC sometimes has better NMI scores for the segment clustering, in almost all situations it finds significantly more segment clusters. The impact on NMI of this finer clustering is explored further in Figure 5.8. We can see that the GMC has higher homogeneity, and lower completeness than the

Figure 5.6 – The segment clusters corresponding to Figure 5.5 shown independently of the image clusters. The rows are random samples of images with a segment cluster present within them. Only the 12 most frequent segment clusters are shown.

SCM variants. Combined with the knowledge of the GMC having more clusters, we can say that it partitions the feature space more finely, leading to many more small homogeneous clusters per class than the SCM. That is, it requires many more clusters to achieve the same performance than the SCM variants, over-clustering and arguably making for a more trivial, less useful, solution.

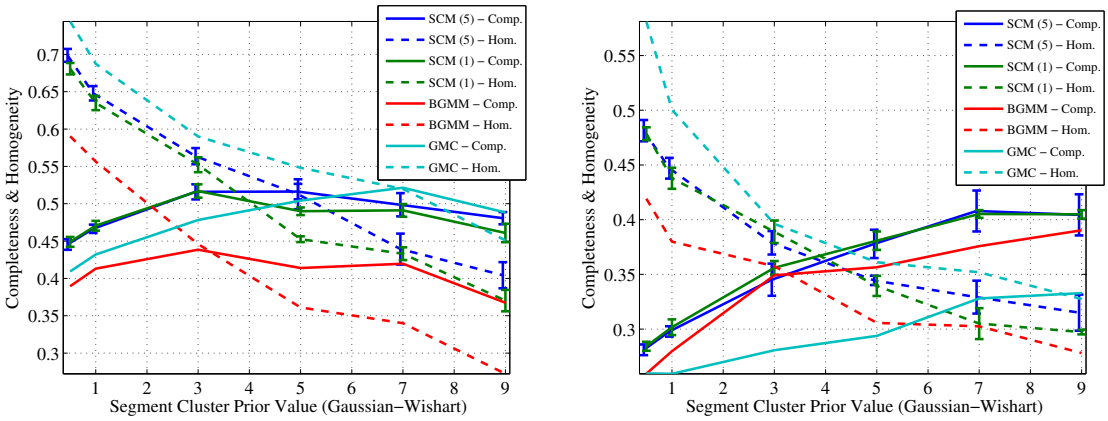
Finally, in Table 5.1 a comparison is made between the BGMM and SCM (1) for *image* clustering performance. In this experiment the BGMM uses the ScSPM descriptors. We can see that for the MSRC dataset the best SCM (1) result is significantly better the best BGMM result. However the reverse is true for the outdoor scenes dataset. In both instances the SCM finds significantly more clusters.



(a) Sample image clusters **(b)** Corresponding segment clusters

Figure 5.7 – A sample SCM (5) result on the outdoor scenes dataset, with $C_{width} = 7$, and $\text{NMI}_i = 0.536$, $\text{NMI}_s = 0.350$, $K = 8$, and $T = 16$. Random image cluster samples are shown in (a) across the rows, and the corresponding segment clusters are shown in (b). This run of the algorithm converges to a strange result with the fourth last image cluster. Only very occasionally with this dataset was this observed.

In this experiment we can see that the SCM requires fewer segment clusters than the GMC to achieve mostly as good or better results, and in less time. Also, we can see the importance of modelling some form of context, since the BGMM performs the worst out of all models tested for clustering segments. The story is less clear for image clustering performance between the SCM and the BGMM using ScSPM descriptors.



(a) MSRC-2 Dataset **(b)** Outdoor Scenes Dataset

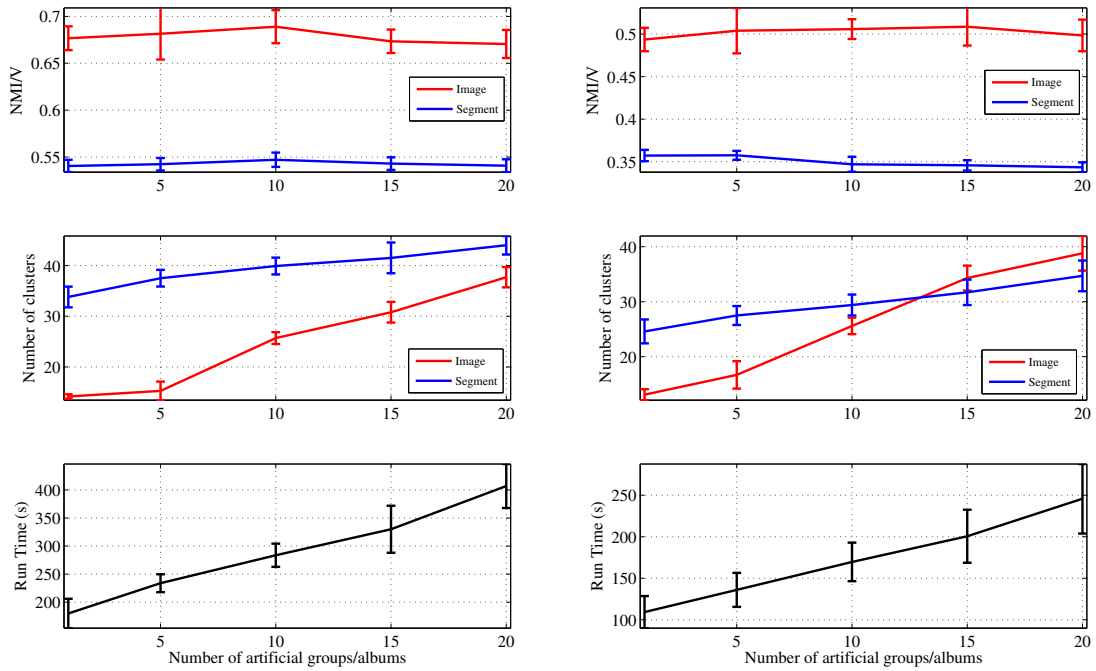
Figure 5.8 – Looking at the components of V-measure (NMI), homogeneity and completeness, for the *segments*. We can see the GMC generally has a higher homogeneity, but lower completeness. Indicating that it is relying on a finer grained clustering to achieve the same performance as the SCM algorithms.

5.6.2 Number of Albums/Groups

In the previous experiment it was unclear as to whether modelling the context inherent within groups benefited clustering of both imagery and segments for the SCM. This section will vary the number of groups used in the SCM in order to more fully explore this relationship. Results are presented in Figure 5.9.

We can see from Figure 5.9 that as the number of artificial groups increases, NMI stays relatively constant. Furthermore, the number of image clusters tends to increase without noticeable bounds for the number of groups tested. The number of segment clusters also tended to increase, but at a lower rate. Both these increases lead to longer runtime. For the image clusters, this is quite different behaviour to the GMC when applied to image clustering from Chapter 4.

The reasons for this behaviour may be that a Multinomial distribution is used to represent image clusters, rather than the Gaussian clusters used in Chapter 4. The idea of different groups disambiguating heavily overlapping clusters in feature space may not apply as well to a Multinomial cluster representation. Also, compared to high-dimensional Gaussians, Multinomial clusters may not contribute enough to the free



(a) MSRC-2 Dataset **(b)** Outdoor Scenes Datset

Figure 5.9 – Results of changing the number of artificial groups changes the SCM results. A prior cluster width of $C_{width} = 1$ was used in both cases.

energy complexity penalty to regulate the number of image clusters in this situation, and with these choices of priors.

From these experiments, it can be seen that using group context, or treating multiple datasets or albums in a distinct manner in the SCM does not seem to benefit the clustering solutions. However, it is reasonable to believe that this is a result of the Multinomial image representation.

5.6.3 Model Prior

The choice of model prior distribution for the image cluster weights and the image cluster parameters (segment cluster weights) may have a large effect on clustering performance. This section performs the same experiments as Section 5.6.1, but only

using variants of SCM (1) with different prior distribution combinations:

1. Generalised Dirichlet on class weights, Dirichlet on cluster weights, referred to as "G-D".
2. Symmetric Dirichlet prior on class and cluster weights ("D-D").
3. Generalised Dirichlet prior on class and cluster weights ("G-G").

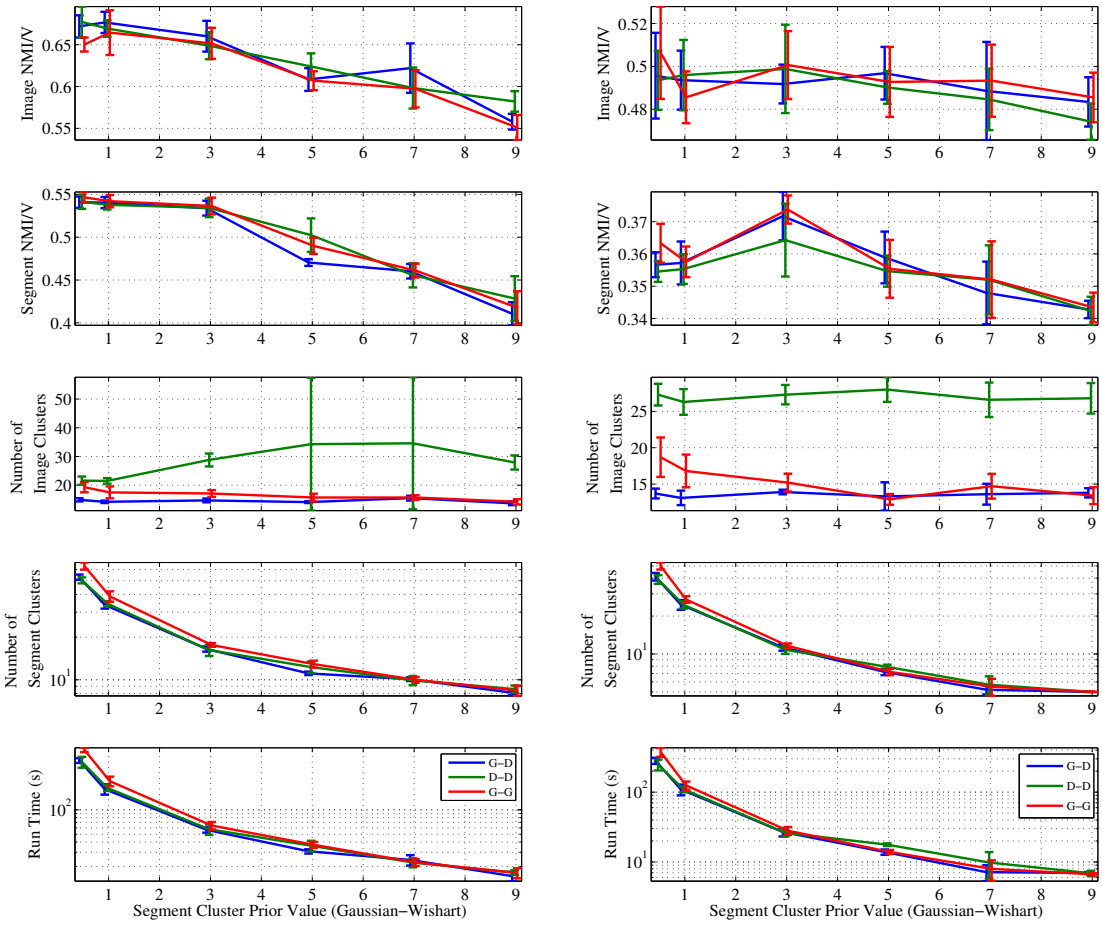
The MSRC-2 and Outdoor Scenes datasets are used again with the same image features, and the results for performance are summarised in Figure 5.10. We can see that they all achieve *remarkably* similar NMI scores for both image and segment clusters. The only attribute that seems to differentiate them is the number of image clusters found. In this case, the D-D prior combination tends to find far more. This is attributable to the generalised Dirichlet having a larger impact on free energy model complexity penalties, since it has twice the number of parameters as the Dirichlet. The generalised Dirichlet is also slightly more flexible than the Dirichlet in modelling the covariance between image clusters, which may have a small impact [126].

The completeness and homogeneity components of NMI for the *image clusters* are plotted in Figure 5.11. We can see that the D-D variant obtains its NMI score through finding many small homogeneous clusters to describe a single image class. Again, this solution seems somewhat more trivial than having more even completeness and homogeneity components.

Interestingly, all variants appear to perform very similarly when clustering segments. This may be attributable to the high-dimensional Gaussian segment clusters having strongly negative log-likelihoods, and high complexity penalties relative to Categorical/Multinomial distributions. The corresponding terms may dominate the priors and contributions of the weights, $\boldsymbol{\beta}_t$, in the expectation step, Equation 5.9, and free energy objective function. This is also something that was observed in Chapter 4 when trialling different weight prior distributions with the GMC. It can be concluded that the original SCM formulation is an appropriate compromise between modelling performance, and model simplicity.

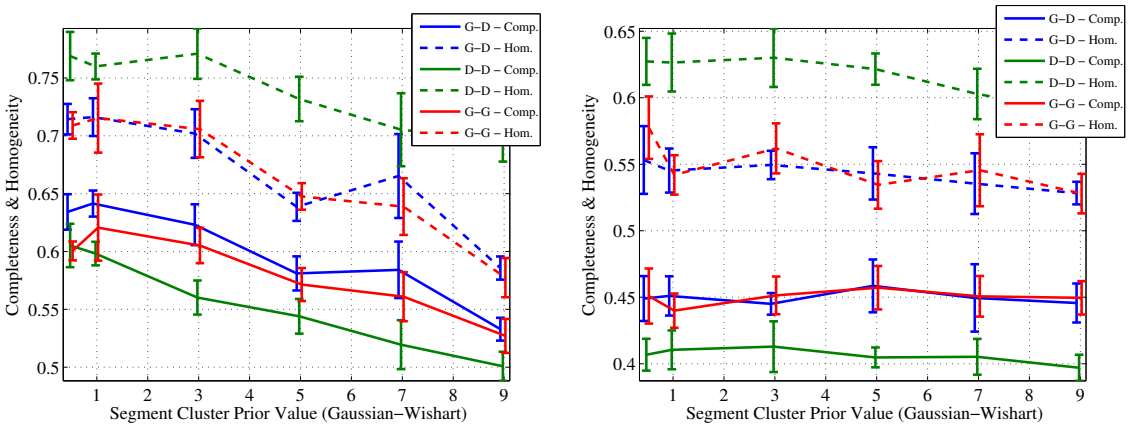
5.6.4 Case study on a Scientific Dataset

For the final experiment a dataset obtained from stereo cameras on an AUV was used. The dataset has approximately 2800 images selected randomly from 5 survey dives, which are the groups. These 5 survey dives are a subset of labelled images from the 12 dives used in Chapter 4. Only the colour images of the stereo pair were



(a) MSRC-2 Dataset **(b)** Outdoor Scenes Datset

Figure 5.10 – The effect of using different priors on clustering results. G-D means a Generalised Dirichlet has been used on the image cluster weights, and a Dirichlet on the segment cluster weights/image clusters, D-D mean a Dirichlet has been used on both image and segment cluster weights, and so forth.



(a) MSRC-2 Dataset **(b)** Outdoor Scenes Datset

Figure 5.11 – Looking at the components of V-measure (NMI), homogeneity and completeness, for the *images*. We can see the Dirichlet-Dirichlet prior combination generally has a higher homogeneity, but lower completeness. This indicates that it is relying on a finer grained clustering to achieve the same performance of as the Generalised Dirichlet priors.

used. The surveys were conducted over rocky reefs near the Tasman National Park on the East coast of Tasmania, Australia [125]. The images are of various habitats on the sea floor and taken at a target altitude of two metres. The original images were 1360 × 1024 pixels in size, but were reduced to 1260 × 924 after correcting for lens distortion, and cropping. The images were further reduced in size to 320 × 235 pixels to make segmentation and feature extraction feasible.

This dataset had 9 image classes; sand/reef interface, low relief reef, coarse sand,

Figure 5.12 – Some exemplar images of the AUV dataset ground truth classes.

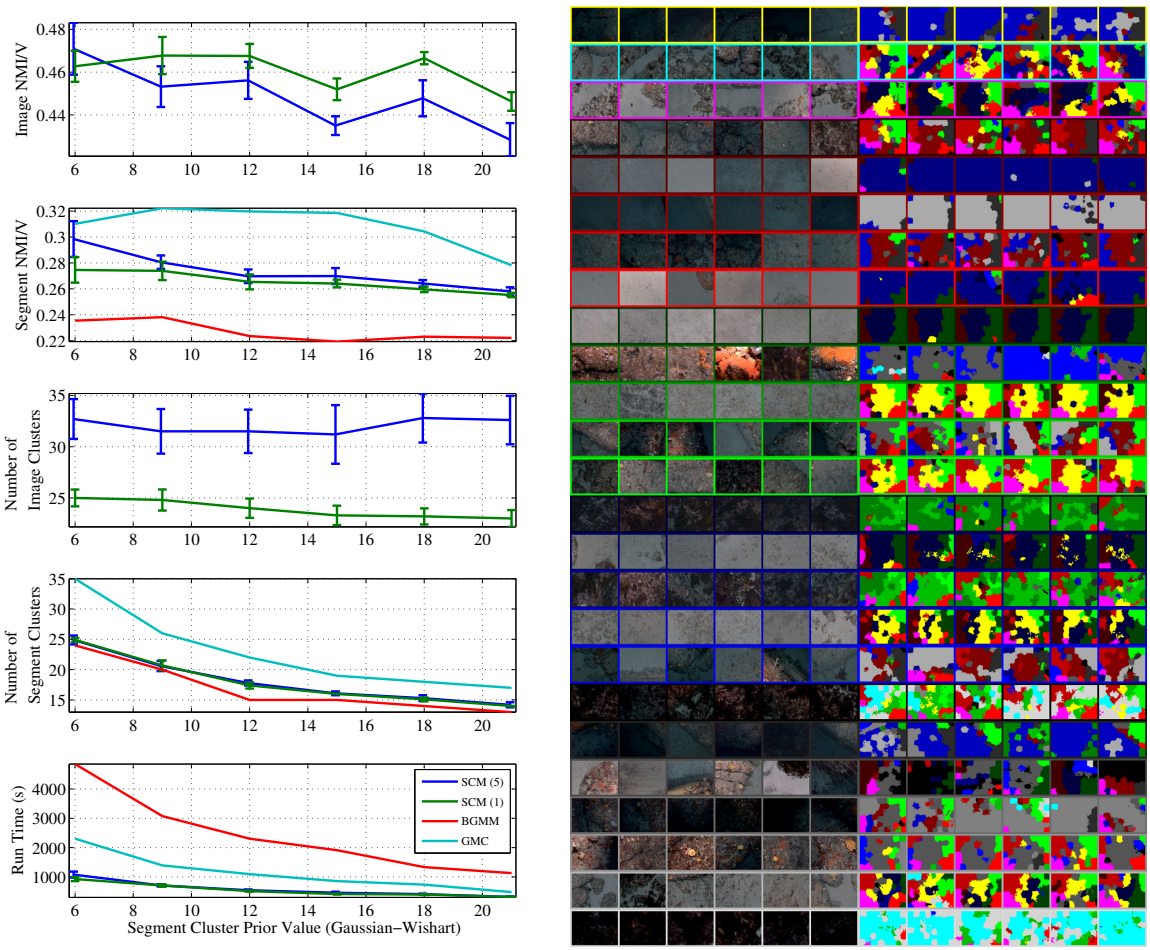
patch reef, fine sand, screw shell rubble ($> 50\%$), screw shell rubble ($< 50\%$), high relief reef, and Ecklonia (Kelp). Some exemplar images are shown in Figure 5.12.

This dataset did not have segment labels, but rather had coral point count (CPC) labels, which are 50 random points labelled according to the object they cover. Furthermore, only 439 images had these CPC labels, and some of these points were lost after the image un-distortion and cropping procedure. There were two classes which were uninformative and relatively large; the biological matrix class (4597 points, 35.7%), and the unknown class (607 points, 4.7%). The sand class also made up a large proportion of these points (5060 points, 39.3%), and accounted for over 65% of the remaining dataset once the two uninformative classes were removed. To avoid rewarding trivial clustering solutions the sand class was also removed, leaving the labels summarised in Table 5.2. A segment was assigned a label according to the mode of CPC labels appearing within the segment.

Patch DC component removal and contrast normalisation were found to greatly improve performance on this dataset, most likely because of the large illumination variation in the imagery. 5×5 pixel patches were optimal, and 200,000 patches were used to train an ICA dictionary with 50 filters.

A few pathologies were present within this dataset, which made it more challenging for clustering. Firstly, different light wavelengths attenuate at different rates within the water column, resulting in red hues close to the camera, and blue far away. Secondly, the air-glass-water interface of the camera housing introduced more distortion and chromatic aberration towards the image boundaries than could be corrected with the camera calibration models used. Unfortunately compensating for these problems are still open research questions, and so were not dealt with here. These pathologies led to several clusters being found that captured the red-blue attenuation and texture-aberration distortion effects.

Results of this experiment, as well as a sample clustering result from the SCM (1), are presented in Figure 5.13 and Figure 5.14. Overall the NMI figures are quite low, despite fairly homogeneous looking clusters in Figure 5.13b. This can be partially explained by some of the semantic content from the labels being difficult to observe



(a) Clustering Results **(b)** Sample image clusters **(c)** Corresp. segment clusters

Figure 5.13 – SCM sample clustering results on the AUV dataset, (a). Sample image clusters (b), and the corresponding segment clusters (c), from the SCM (1) with $C_{width} = 9$, $\text{NMI}_i = 0.466$, $\text{NMI}_s = 0.274$, $K = 21$ and $T = 25$.

directly in the images. For example, the difference between patch and low relief relies on neighbouring images to determine the extent of the reef. If it is "small" it is a patch reef. Furthermore, the SCM variants have arguably over-clustered this dataset. Interestingly, the SCM (1) convincingly outperforms the SCM (5) variant here in image clustering, for most C_{width} values. Though, these values are worse than the GMC image clustering results from Chapter 4 for the full 100,000 image dataset.

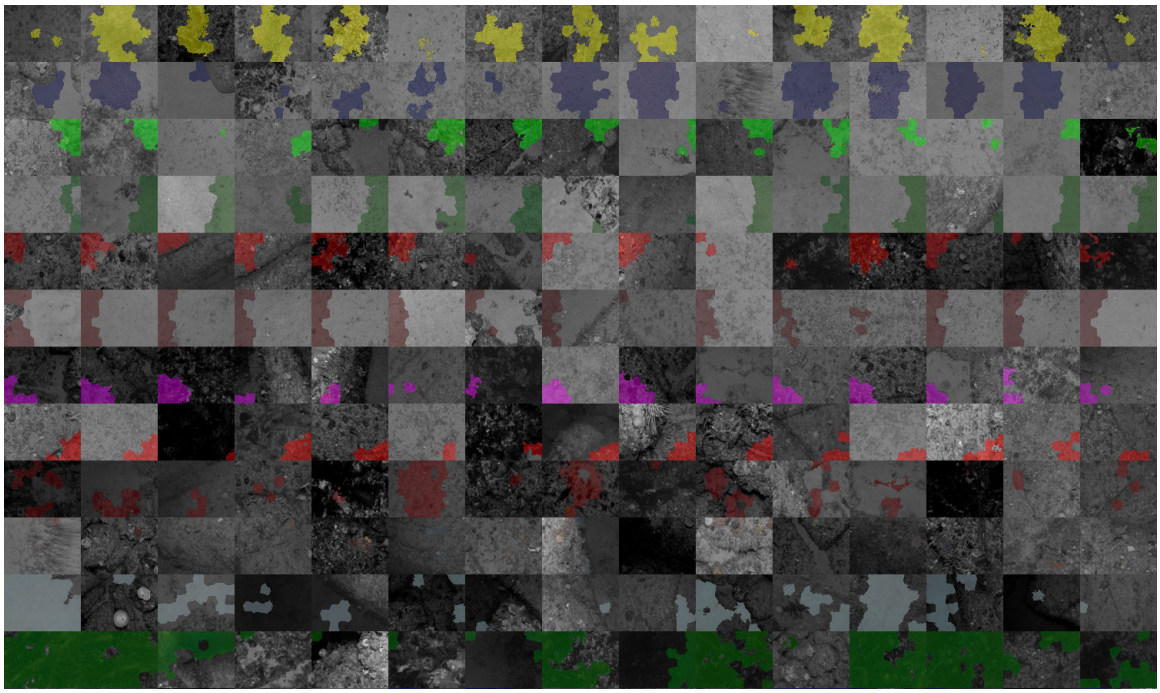


Figure 5.14 – The segment clusters corresponding to Figure 5.13 shown independently of the image clusters. The camera distortion influence on the segment clusters is fairly apparent in some of these clusters. Again, the 12 most frequent segment clusters are shown.

When the GMC is applied to this 2800 image dataset, it can achieve an NMI in excess of 0.5 (shown in the next chapter).

Both SCM variants perform similarly for segment clustering, and are both soundly beaten by the GMC, which also consistently finds more clusters. Unlike Section 5.6.1, the completeness and homogeneity components of the GMC segment clusters are both always higher than those of the SCM variants in this experiment. However, by far the fastest algorithms on this dataset are the SCM variants. It is very interesting to note that the BGMM takes far longer to find fewer and worse clusters than the other models. This exemplifies that modelling context in some form is beneficial for clustering.

While we saw in Section 5.6.3 that using a generalised Dirichlet prior helped to control the number of image clusters found by the SCM, in Figure 5.13 the number is still high (around 30). It was found that modifying the image cluster prior hyper-parameter,

ϕ, could also help control the number of image clusters found, as demonstrated in Figure 5.15. Here ϕ for the SCM (1), (5) and symmetric Dirichlet (D-D) variants was changed, while holding $C_{width} = 6$ constant. As before, image and segment clustering performance is similar between all of the variants, but the symmetric Dirichlet variant consistently found more image clusters, and quite often hit the T_{trunc} value for high values of ϕ. The generalised Dirichlet variants responded quite well different values of ϕ.

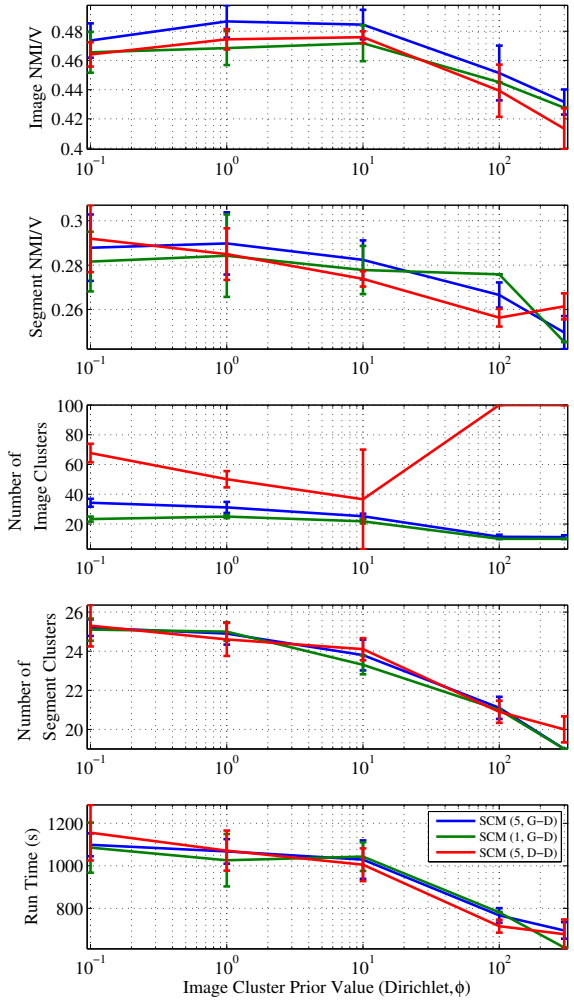


Figure 5.15 – Here the SCM image cluster prior, ϕ, is changed to show how it affects clustering results of the SCM variants. For all of these experiments, $C_{width} = 6$.

The prior parameter, ϕ, essentially controls how many different objects (segment clusters) we expect to be in a particular scene-type (image cluster) a-priori. For low values of ϕ, we would expect only a few objects within each scene-type (it is a sparse prior), i.e. we expect z_{jin} to only take a few values of k for a particular scene-type, t. Whereas for high values of ϕ, we would expect many more objects to exist in each scene-type. So, what we observe in Figure 5.15, is that for low values of ϕ, more image clusters are required to represent all possible object-types, k, since only a few can exist in a scene-type. However, for high values of ϕ, more objects can exist in fewer scene-types.

In Figure 5.13c and Figure 5.14 we can observe how the image distortion affects the segment clustering results. In a few of the more uniform image clusters (such as those corresponding to the sand and rubble classes), the images look as if they have been segmented into portions consistent with the distortion pattern of the images. For example the tenth image cluster from the top in Figure 5.13c, which corresponds to sand, very consistently shows this problem. Something else that was found to be unique to this dataset was that any other patch size apart from 5×5 pixels resulted in a drastic fall-off of NMI, to the order of 7% or more for both image and segment clusters – even with ± 1 pixel variation. This fact, combined with the distortion artefacts, suggests that a more invariant image representation would be desirable in this situation. Furthermore, it would be useful to extend these models to make use of the CPC labels in a semi-supervised manner, which would make up for some of the semantic content missing in the visual data alone.

5.6.5 Case Study on a Photo Collection

For the final experiment the photo albums dataset from Section 4.6.4 was used. This dataset consists of 12 photo albums, and was constructed based on holidays of the author. Approximately 2200 photos are from the author, and 8100 images from *Flickr* (creative commons), downloaded from the same locations based on relevance.

Unfortunately, unlike the experiment in Section 4.6.4, the likelihood of this model has

no closed form solution because of the interaction between the image and segment labels, $\mathbf{Y}$ and $\mathbf{Z}$. So, this experiment is purely qualitative in nature.

There are 10,324 images in total with a maximum dimension of 320 pixels, and a total of 524,187 super-pixels were extracted from these images (approximately 50 per image). The ICA dictionary for the segment observations, $\mathbf{x}_{jin}$, was learned from 200,000 random patches of 5×5 pixels, and like the other non-underwater datasets, DC component removal and contrast normalisation did not help results. Again PCA whitening was used on the final pooled segment descriptors, and dimensionality was reduced to $D_2 = 15$. The segment features take around 1 second to calculate per image.

This dataset is less constrained in terms of the diversity of scene types, and their inherent proportions than the other datasets, and so poses a good challenge for unsupervised scene analysis. Multi-threading was used in this example, and kept two cores almost constantly under 100% load. A sample SCM result is shown in Figure 5.16, which used $\phi = 300$ and $C_{width} = 20$. This took 1336 seconds (22 min, 16 sec) and found $T = 23$ image clusters, and $K = 17$ segment clusters. Also shown are the top seven tags by frequency in each of the image clusters.

While some of the image clusters in this result are reasonably uniform (like those involving water and plants), the majority do not look as cohesive as those found by the VDP and GMC in Section 4.6.4. For instance, the spatial layout of the SCM's clusters are not as consistent as those of the VDP or GMC in the last chapter. This is because the ScSPM image descriptors capture crude spatial layout, whereas the ICA descriptors do not. However, there are a few clusters that look more consistent in terms of colour (or lack of), like the 3rd from the bottom.

5.7 Summary

Taking advantage of group or album context did not seem to consistently help the image clustering result found by the SCM. In some situations it even led to worse

Figure 5.16 – Sample SCM clustering results on the photo album dataset. Random images from each cluster are shown, with the corresponding segment clusters. The seven most frequent tags per image cluster are also shown. There are 23 image clusters, and 17 segment clusters, see text for more details.

results. However it does clearly benefit the GMC when used for image clustering in Chapter 4, and here at the segment level. This suggests that the choice of clustering distribution heavily influences whether or not sharing clusters between groups will aid inference. It was shown in Chapter 4 that as the number of groups increased, the number of Gaussian image clusters found would not change significantly in the GMC, but the clustering solution usually improved. However it was found here that as the number of groups increased, the number of Multinomial image clusters in the SCM increased rapidly, while the clustering solution did not change significantly in quality.

The generalised Dirichlet prior on the group weights, $\boldsymbol{\pi}_j$, tended to penalise "over-clustering" more than a Dirichlet, without sacrificing image clustering quality. It is also slightly more flexible than the Dirichlet in modelling correlation between image clusters. However, as in Chapter 4, a generalised Dirichlet prior over segment cluster weights/image cluster parameters, $\boldsymbol{\beta}_t$, did not affect the segment clustering greatly compared to the original use of a Dirichlet. This is probably because of the strong effect of the Gaussian log-likelihood and its complexity penalty in the free energy objective function, as opposed to those of a Multinomial. It was found that a good combination of priors for the SCM is a generalised Dirichlet over image cluster weights, and a Dirichlet over image cluster parameters.

The SCM tended to under-perform the BGMM and GMC for clustering images (when they used ScSPM image descriptors). This is because the ICA features and Multinomial image representation were deficient in some aspects for unsupervised applications. For example, the Multinomial "bag-of-segments" representation, and ICA descriptors are unable to model spatial layout, unlike the ScSPM descriptors.

Despite the apparent weakness in using a Multinomial distribution to represent image clusters, the experiments conclusively show that using various forms of context help achieve better segment clustering solutions, and often in less time. Image level context (GMC) seems to often lead to more, but potentially better segment clusters[4]. Image cluster context and object co-occurrence (SCM) performed much better than not

[4]Subsequently, it has been found that the GMC does not perform as well as the SCM when mean-shift [32] segments are used.

taking context into account (BGMM), and was sometimes better than the GMC, and led to the fastest run-times in all experiments. This fast runtime is because the SCM has less weight distributions to update in its maximisation step than the GMC (one per image cluster, as opposed to one per image), and it also usually found less segment clusters. Additionally, the SCM could cluster on multiple levels simultaneously while the other models tested could not. Like in Chapter 4, the segment clustering result is improved in part because taking into account context allows for multiple views of the observations in feature space. This is opposed to clustering all observations as if they were in one bag, which is what traditional mixture models do.

One of the weaknesses of the segment representation used is the dependence of the object discovery performance on the parameters chosen for ICA and the image segmentation algorithm. It would be interesting to try the SCM on multiple-segmentations like in [95]. However, having multiple segmentations for each image, and thus multiple versions of the same object, violates the assumptions of the SCM in that each image is a distribution of objects (with each object only being represented once). Violating this assumption may lead to poorer image clustering performance, and hence poor object discovery. This may be an interesting avenue to pursue for future work.

As future work it would be interesting to test if a power-law distribution on group weights, $\boldsymbol{\pi}_j$, such as the Pitman-Yor process [89], could improve clustering results in the natural datasets, such as the AUV dataset. It would also be useful to find more suitable methods for representing image clusters in a simultaneous clustering model. A logistic Normal prior with a full covariance matrix, as used in [16], over the Multinomial parameters may more heavily penalise model complexity, though this is a non-conjugate prior, and would also not admit inference of the number of segment clusters. The logistic Normal does model correlation between dimensions, which in this case would also model object (segment cluster) pair-wise co-occurrence. Alternatively, it would be interesting to see if applying a HDP [109] to this problem can improve inference. This model would not find image clusters, but it can be made to model image and album context for segments (extending the GMC), as opposed to image cluster and album context in the SCM.

Image features that are more invariant to distortion and colour inconsistencies would also be desirable. Some invariance could potentially be introduced by using deeper architectures, i.e. have multiple ICA layers, with pooling between each in a similar fashion to [20]. Multi-scale features may also help in this regard.

Another useful avenue of research may be augmenting these models to incorporate "noisy" or potentially incorrect labels and annotations at multiple levels. This has been successfully achieved at the image "tag" or object level by Li et al. [72]. This would have been particularly useful on the AUV dataset, where the original image labels (not used), and some of the CPC labels are definitely incorrect.

Algorithm 5.1: The SCM greedy model selection heuristic

Data: Observations $\mathbf{X}$
Result: Probabilistic assignments $q(\mathbf{Y})$, $q(\mathbf{Z})$ and $\{\tilde{\mathbf{a}}, \tilde{\mathbf{b}}, \tilde{\phi}, \tilde{\eta}, \tilde{\boldsymbol{\nu}}\}$

$\{\mathbf{a}, \mathbf{b}, \phi, \eta, \boldsymbol{\nu}\} \leftarrow$ `CreatePriors()`;
$q(\mathbf{Y}) \leftarrow$ `RandomLabels(`$T_{trunc} = 100$`)`;
$q(\mathbf{Z}) \leftarrow \{\{\mathbf{1}\}_{i=1}^{I_j}\}_{j=1}^{J}$; `// initialises with` $K = 1$
splittally $\leftarrow \{0\}_{k=1}^{K}$;

repeat
  $q(\mathbf{Y}), q(\mathbf{Z}), \{\tilde{\mathbf{a}}, \tilde{\mathbf{b}}, \tilde{\phi}, \tilde{\eta}, \tilde{\boldsymbol{\nu}}\}, \mathcal{F} \leftarrow$ `VarBayes(`$\mathbf{X}, q(\mathbf{Y}), q(\mathbf{Z}), \{\mathbf{a}, \mathbf{b}, \phi, \eta, \boldsymbol{\nu}\}$`)`;
  splitorder $\leftarrow$ `GreedySorter(`$\mathbf{X}, q(\mathbf{Z}), \eta, \boldsymbol{\nu}$, splittally`)`; `// this is a 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(`$\mathbf{X}_{split,k}$`)`;
    $q(\mathbf{Z}_{split,k}) \leftarrow$ `VarBayes(`$\mathbf{X}_{split,k}, \{\mathbf{1}\}_{j=1}^{J}, q(\mathbf{Z}_{split,k}), \{\mathbf{a}, \mathbf{b}, \phi, \eta, \boldsymbol{\nu}\}$`)` ;
     `// refine`
    $q(\mathbf{Z}_{aug,k}) \leftarrow$ `AugmentLabels(`$q(\mathbf{Z}), q(\mathbf{Z}_{split,k})$`)`; `// add in split labels`
    $\mathbb{E}[\mathcal{F}_{split,k}] \leftarrow$ `VarBayes(`$\mathbf{X}, q(\mathbf{Y}), q(\mathbf{Z}_{aug,k}), \{\mathbf{a}, \mathbf{b}, \phi, \eta, \boldsymbol{\nu}\}$`)`; `// 1 iteration`
    if $\mathcal{F} > \mathbb{E}[\mathcal{F}_{split,k}]$ **then**
      $q(\mathbf{Z}) \leftarrow q(\mathbf{Z}_{aug,k})$;
      $\text{splittally}_k \leftarrow 0$;
      $\text{splittally}_{K+1} \leftarrow 0$;
      foundsplit $\leftarrow$ *true*;
      break;
    else
      $\text{splittally}_k \leftarrow \text{splittally}_k + 1$;
      foundsplit $\leftarrow$ *false*;
until foundsplit = *false*;

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

Table 5.1 – Comparison of clustering images using the BGMM with ScSPM image descriptors, and the SCM with one group. The subscripts i and s means that the C_{width} prior has been applied to image and segment Gaussian clusters respectively.

	MSRC		Outdoor	
Algorithm	**NMI**	**T**	**NMI**	**T**
BGMM+ScSPM	0.5734 ($C_{width,i} = 0.03$)	9	0.5677 ($C_{width,i} = 0.05$)	4
SCM (1)	0.6767 ($C_{width,s} = 1$)	14.2	0.4968 ($C_{width,s} = 5$)	13.3

Table 5.2 – Summary of labels used for validation.

(a) Image Labels

Name	Count	Percent
Sand/reef interface	110	3.89
Low relief reef	438	15.50
Coarse sand	370	13.10
Patch reef	153	5.42
Sand	83	2.94
Screw shells	264	9.35
Few screw shells	275	9.73
High relief reef	750	26.55
Ecklonia (kelp)	382	13.52

(b) CPC labels

Name	Count	Percent
Coral	203	7.75
Ascidians	1	0.04
Bryozoans	96	3.67
Echinodermata	15	0.57
Fish	10	0.38
Mollusca	244	9.32
Macroalgae	15	0.57
Red Macroalgae	663	25.32
Brown Macroalgae	583	22.26
Sponges	522	19.93
Biological Rubble	258	9.85
Bare Rock	9	0.34

Chapter 6

Clustering Observations of Images and Image Parts

It was seen in the last chapter that jointly clustering image segments or super-pixels into unsupervised "objects" while clustering images based on the proportions of these objects within them, improved performance and run-time over just clustering image segments with *no* context (i.e. using regular mixture models). Unfortunately the model presented for this task, the simultaneous clustering model (SCM), had a number of limitations. Firstly, it did not seem to consistently take advantage of the context inherent in groups or albums as explored in Chapter 4. It was hypothesised that this was because of the inherent multinomial representation of image clusters used. The SCM also tended to need a larger number of image clusters than the other models for good performance. The grouped mixtures clustering model (GMC) exhibited none of this behaviour in Chapter 4 for clustering images. In this chapter, the SCM and GMC are combined into one unified model for representing images. This model exhibits none of the limitations of the SCM, and retains the GMC's ability to take advantage of groups or albums. Image segment or super-pixel, and whole image features are used by this model. This approach is inspired by supervised models in the literature that provide scene label training data for enhancing object recognition tasks – but naturally this is applied in an unsupervised setting.

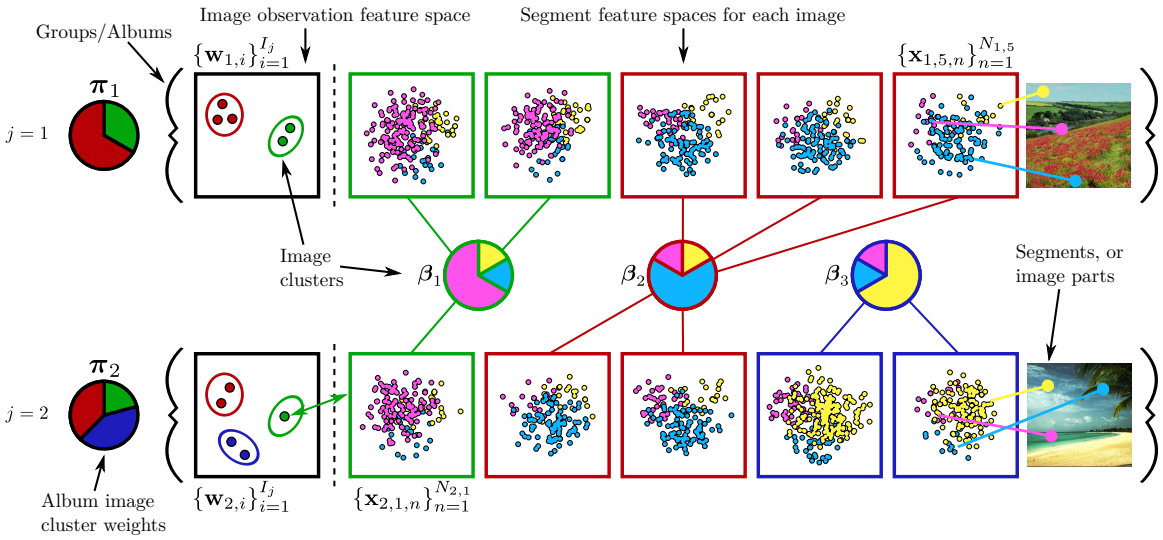


Figure 6.1 – Demonstration of clustering multiple observation sources in groups. Here there are image observations (red, green blue points), $\mathbf{w}_{ji}$, and segment or super-pixel observations (yellow, magenta and cyan points), $\mathbf{x}_{jin}$. Each coloured square represents an image's super-pixels in feature space. The images are indexed by i. The black squares are the feature spaces of the image observations per group/album (curly braces), indexed by j. Segments are clustered into "objects" (cyan $\approx$ plant, magenta $\approx$ water and yellow $\approx$ sky), and are shared between images and groups. Images with a similar proportion and co-occurrence of segment clusters, $\boldsymbol{\beta}_t$, form part of the image cluster descriptions (red, green and blue squares), along with Gaussian clusters of image observations (ellipses). Groups can be described by the proportions of image clusters within them, $\boldsymbol{\pi}_j$. As in Chapter 4 there are multiple views of the observations in feature space, which simplifies and improves inference.

6.1 Introduction

In this chapter a model is developed that uses a joint representation of images clusters for unsupervised image understanding. This model can use whole image or scene observations like those used by the GMC in Chapter 4. It also uses observations of segments or super-pixels like the SCM in Chapter 5. A diagram of this image representation is presented in Figure 6.1.

The intention of this chapter is to essentially overcome some of the limitations of the SCM from the last chapter, while retaining the useful description of an image as a combination of "objects". The inspiration for this work comes from the supervised image understanding literature, where it has been established by works such as

Choi et al. [31], Torralba and Oliva [114], Torralba et al. [115, 116] that classifying an image into a scene type can be used to improve the (supervised) recognition of objects. Similarly, it has been shown that using global image cues can also aid in object discovery [68, 97]. Much of this work is in-turn inspired by research on the human visual cortex [85], which uses global visual features to recognise a scene-type, or get the "gist" of the scene, without explicitly registering the objects within that scene. It has been demonstrated that this scene recognition provides context that aids the recognition of objects, which otherwise may be difficult to recognise in isolation. Analogously, the GMC, when used to cluster images, may be seen as using only the "gist" of the scene. Whereas the SCM has to create a representation of a scene type from the proportions of objects within a images. So the objective of this chapter is to combine these different image representations into one unified model for unsupervised, annotation-less, image understanding. The contribution of this chapter is in formulating such a model, and experimentally evaluating its performance.

In Section 6.2 the proposed model is introduced, and its generative process explained. Section 6.3 outlines the variational Bayes (VB) learning algorithm for this model. Experiments are performed in Section 6.5 that use datasets from the previous two chapters. Finally a summary and discussion is provided in Section 6.6.

6.2 Clustering Observations of Images and Image Parts in Groups

In this section the model for simultaneously clustering observations of super-pixels and images is introduced. It is referred to as the multiple-source clustering model (MCM), and is very similar to the SCM from the previous chapter. The major difference is now there is a joint Multinomial-Exponential family mixture representation of image clusters. The Multinomial distribution is inherited from the SCM and represents the segment cluster proportions per image cluster. The Exponential family clusters are used to describe the image observations, like in the GMC. These distributions

share a common image label. It is hoped that this joint representation of images will remedy some of the limitations of the SCM. These limitations are its propensity to "over-cluster" images in large datasets to achieve good performance, also the SCM demonstrated that in many cases it did not take advantage of groups or albums.

Following Figure 6.1, image observations $\mathbf{w}_{ji} \in \mathbb{R}^{D_1}$, and image super-pixels or segments $\mathbf{x}_{jin} \in \mathbb{R}^{D_2}$, are assumed to be arranged in the following manner:

- There are N_{ji} segments, $\mathbf{X}_{ji} = \{\mathbf{x}_{jin}\}_{n=1}^{N_{ji}}$, in each image, which is described by $\mathbf{w}_{ji}$.
- There are I_j images in a group, or "album", $\mathbf{W}_j = \{\mathbf{w}_{ji}\}_{i=1}^{I_j}$ and $\mathbf{X}_j = \{\mathbf{X}_{ji}\}_{i=1}^{I_j}$.
- There are J groups in the whole dataset, $\mathbf{W} = \{\mathbf{W}_j\}_{j=1}^{J}$ and $\mathbf{X} = \{\mathbf{X}_j\}_{j=1}^{J}$.

The aim is to discover K segment clusters, parameterised by $\boldsymbol{\Theta} = \{\theta_k\}_{k=1}^{K}$, shared between all of the images, and T image clusters, parameterised by $\mathbf{B} = \{\boldsymbol{\beta}_t\}_{t=1}^{T}$ *and* $\boldsymbol{\Sigma} = \{\sigma_t\}_{t=1}^{T}$, shared between the groups/albums. The tth image cluster parameters include proportions of segment clusters, $\boldsymbol{\beta}_t = [\beta_{t1}, \ldots, \beta_{tk}, \ldots, \beta_{tK}]$, where $\beta_{tk} \in [0, 1]$ and $\sum_k \beta_{tk} = 1$. Furthermore, the jth group or album is described by the proportions of the image clusters within it, $\boldsymbol{\pi}_j = [\pi_{j1}, \ldots, \pi_{jt}, \ldots, \pi_{jT}]$, again $\pi_{jt} \in [0, 1]$ and $\sum_t \pi_{jt} = 1$. Latent auxiliary variables are used for assigning images to image clusters, y_{ji}, and segments to segment clusters, z_{jin}. Once the cluster parameters have been drawn; $\theta_k \sim p(\eta, \boldsymbol{\nu})\, \forall k$, $\sigma_t \sim p(\gamma, \boldsymbol{\delta})\, \forall t$ and $\boldsymbol{\beta}_t \sim \mathrm{Dir}(\phi)\, \forall t$[1], the following generative process for this model is assumed for a group, j,

1. Draw group mixture weights, $\boldsymbol{\pi}_j \sim \mathrm{GDir}(\mathbf{a}, \mathbf{b})$.
2. For each of the I_j images in group j,
   (a) Choose an image cluster, $y_{ji} \sim \mathrm{Cat}(\boldsymbol{\pi}_j)$, where $y_{ji} \in \{1, \ldots, T\}$.
   (b) Draw an image observation, $\mathbf{w}_{ji} \sim p\big(y_{jin}, \boldsymbol{\Sigma}\big)$ from an exponential family distribution with parameters $\boldsymbol{\Sigma}$ indexed by the label y_{ji}.

[1] A scalar hyper-parameter argument in $\mathrm{Dir}(\cdot)$ or $\mathrm{GDir}(\cdot, \cdot)$ means the same value is used for all hyper-parameters.

(c) For each of the N_{ji} segments in image ji,

i. Choose a segment cluster, $z_{jin} \sim p\big(y_{ji}, \mathbf{B}\big)$, from a Categorical distribution with parameters $\mathbf{B}$ indexed by y_{ji}, where $z_{jin} \in \{1, \ldots, K\}$.

ii. Draw a segment observation, $\mathbf{x}_{jin} \sim p(z_{jin}, \boldsymbol{\Theta})$, from an exponential family distribution with parameters $\boldsymbol{\Theta}$ indexed by the label z_{jin}.

The collection of all of the group mixture weights is termed $\boldsymbol{\Pi} = \{\boldsymbol{\pi}_j\}_{j=1}^{J}$. The graphical model for this generative process is given in Figure 6.2, and the corresponding joint distribution is,

$$p(\mathbf{W}, \mathbf{X}, \mathbf{Y}, \mathbf{Z}, \boldsymbol{\Pi}, \mathbf{B}, \boldsymbol{\Sigma}, \boldsymbol{\Theta} | \mathbf{a}, \mathbf{b}, \phi, \gamma, \boldsymbol{\delta}, \eta, \boldsymbol{\nu}) = \prod_{k=1}^{K} p(\theta_k | \eta, \boldsymbol{\nu}) \times \prod_{t=1}^{T} \mathrm{Dir}(\boldsymbol{\beta}_t | \phi)\, p(\sigma_t | \gamma, \boldsymbol{\delta}) \prod_{j=1}^{J} \mathrm{GDir}(\boldsymbol{\pi}_j | \mathbf{a}, \mathbf{b}) \prod_{i=1}^{Ij} \mathrm{Cat}\big(y_{ji} | \boldsymbol{\pi}_j\big)\, p\big(\mathbf{w}_{ji} | y_{ji}, \boldsymbol{\Sigma}\big) \times \prod_{n=1}^{N_{ji}} p\big(z_{jin} | y_{ji}, \mathbf{B}\big)\, p(\mathbf{x}_{jin} | z_{jin}, \boldsymbol{\Theta})\,. \quad (6.1)$$

The following terms can be further factorised,

$$p\big(\mathbf{w}_{ji} | y_{ji}, \boldsymbol{\Sigma}\big) = \prod_{t=1}^{T} p(\mathbf{w}_{ji} | \sigma_t)^{\mathbf{1}[y_{ji}=t]} \quad (6.2)$$

$$p\big(z_{jin} | y_{ji}, \mathbf{B}\big) = \prod_{t=1}^{T} \mathrm{Cat}(z_{jin} | \boldsymbol{\beta}_t)^{\mathbf{1}[y_{ji}=t]} \quad (6.3)$$

$$p(\mathbf{x}_{jin} | z_{jin}, \boldsymbol{\Theta}) = \prod_{k=1}^{K} p(\mathbf{x}_{jin} | \theta_k)^{\mathbf{1}[z_{jin}=k]} \quad (6.4)$$

Recall that $\mathbf{1}[\cdot]$ is an indicator function that returns 1 when the condition in the brackets is true, and 0 otherwise. A Generalised Dirichlet distribution [36, 126] is used over the group mixture weights, $\mathrm{GDir}(\boldsymbol{\pi}_j | \mathbf{a}, \mathbf{b})$, based on the results of Chapter 5. For more details see Chapter 4 or Chapter 5.

For generality and clarity, both image and segment observations, $\mathbf{w}_{ji}$ and $\mathbf{x}_{jin}$, are assumed to be drawn from any exponential family distribution given a mixture com-

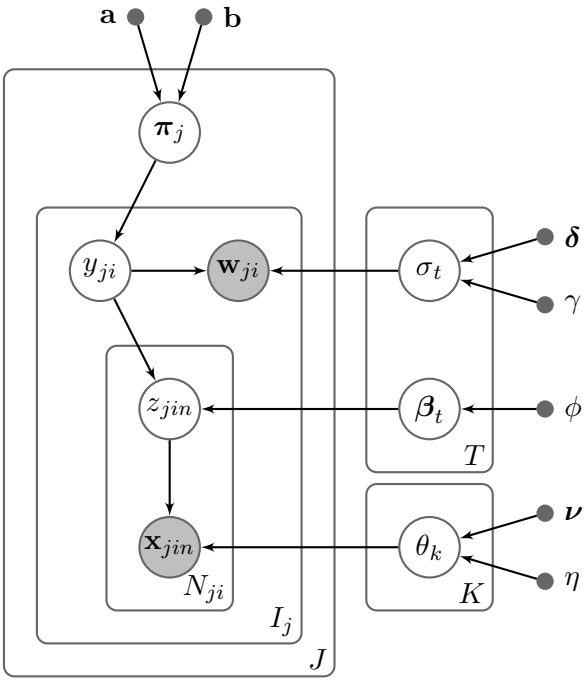


Figure 6.2 – The MCM graphical model. Notice how the top half of this model, i.e. all the nodes involving $\mathbf{w}_{ji}$, is essentially the GMC, and the rest is the SCM. These two models are linked via the image labels, y_{ji}.

ponent (Gaussian distributions are used in the experiments). Their parameters, σ_t and θ_k, are also drawn from conjugate prior distributions. $p(\mathbf{x}_{jin}|\theta_k)$ is the same as presented in Chapter 5, and for $\mathbf{w}_{ji}$,

$$p(\mathbf{w}_{ji}|\sigma_t) = f(\mathbf{w}_{ji})g(\sigma_t)\exp\{\boldsymbol{\phi}(\sigma_t)^\top \mathbf{u}(\mathbf{w}_{ji})\}, \tag{6.5}$$

$$p(\sigma_t|\gamma,\boldsymbol{\delta}) = h(\gamma,\boldsymbol{\delta})g(\theta_k)^\eta \exp\{\boldsymbol{\phi}(\sigma_t)^\top \boldsymbol{\delta}\}. \tag{6.6}$$

Here $g(\sigma_t)$ and $h(\gamma,\boldsymbol{\delta})$ are log-partition or normalisation functions, $\boldsymbol{\phi}(\sigma_t)$ are natural parameters, $\mathbf{u}(\mathbf{w}_{ji})$ are sufficient statistics of the data, and $f(\mathbf{w}_{ji})$ is a function of $\mathbf{w}_{ji}$.

6.3 Variational Bayes for Learning the Model

The derivations are started by approximating the true posterior over the parameters, $p(\mathbf{Y},\mathbf{Z},\boldsymbol{\Pi},\mathbf{B},\boldsymbol{\Sigma},\boldsymbol{\Theta}|\mathbf{X})$, with a family of factorised mean-field approximating distri-

butions,

$$q(\mathbf{Y}, \mathbf{Z}, \mathbf{\Pi}, \mathbf{B}, \mathbf{\Sigma}, \mathbf{\Theta}) = \prod_{k=1}^{K} q(\theta_k) \times \prod_{t=1}^{T} q(\sigma_t)\, q(\boldsymbol{\beta}_t) \times \prod_{j=1}^{J} q(\boldsymbol{\pi}_j) \prod_{i=1}^{I_j} q\big(y_{ji}\big) \prod_{n=1}^{N_{ji}} q(z_{jin})\,. \tag{6.7}$$

Following [9], the negative free energy lower bound is,

$$\begin{aligned}
\mathcal{F}[q(\mathbf{Y}, \mathbf{Z})\,, q(\mathbf{\Pi}, \mathbf{B}, \mathbf{\Sigma}, \mathbf{\Theta})] = \sum_{t=1}^{T} \left\{ \mathbb{E}_{q_{\boldsymbol{\beta}}} \left[\log \frac{q(\boldsymbol{\beta}_t)}{\mathrm{Dir}(\boldsymbol{\beta}_t|\phi)} \right] + \mathbb{E}_{q_\sigma} \left[\log \frac{q(\sigma_t)}{p(\sigma_t|\gamma, \boldsymbol{\delta})} \right] \right\} \\
+ \sum_{k=1}^{K} \mathbb{E}_{q_\theta} \left[\log \frac{q(\theta_k)}{p(\theta_k|\eta, \boldsymbol{\nu})} \right] + \sum_{j=1}^{J} \mathbb{E}_{q_{\boldsymbol{\pi}}} \left[\log \frac{q(\boldsymbol{\pi}_j)}{\mathrm{GDir}(\boldsymbol{\pi}_j|\mathbf{a}, \mathbf{b})} \right] \\
+ \sum_{j=1}^{J} \sum_{i=1}^{Ij} \sum_{n=1}^{N_{ji}} \mathbb{E}_q \left[\log \frac{q\big(y_{ji}\big)\, q(z_{jin})}{\mathrm{Cat}\big(y_{ji}|\boldsymbol{\pi}_j\big)\, p\big(\mathbf{w}_{ji}|y_{ji}, \mathbf{\Sigma}\big)\, p\big(z_{jin}|y_{ji}, \mathbf{B}\big)\, p(\mathbf{x}_{jin}|z_{jin}, \mathbf{\Theta})} \right],
\end{aligned} \tag{6.8}$$

where the last term's expectation is with respect to all of the latent variables and parameters. This last term acts like a data-fitting objective, and the first three terms act as model complexity penalties. Like in Chapter 5 the last term does not simplify down to a "log-likelihood" like term as it does with the Bayesian Gaussian mixture model (BGMM) or GMC. This is because of the interaction between the latent variables, $\mathbf{Y}$ and $\mathbf{Z}$. All of the expectations are given in Appendix A. The learning objective is to minimise this negative free energy.

All of the variational updates in this model are the same as either the GMC or the SCM. Since these two models essentially interact through the image labels, y_{ji}, this is the only variational update that is unique to this model. So, taking $\partial\mathcal{F}/\partial q(\mathbf{Y}) = 0$, while enforcing $\int q(\mathbf{Y}) d\mathbf{Y} = 1$, results in the variational Bayes expectation (VBE) step for the image labels – or the probability an image belongs to an image cluster,

$$q(y_{ji}=t)=\frac{1}{\mathcal{Z}_{y_{ji}}}\exp\left\{\mathbb{E}_{q_\pi}[\log\pi_{jt}]+\sum_{k=1}^{K}\mathbb{E}_{q_\beta}[\log\beta_{tk}]\sum_{n=1}^{N_{ji}}q(z_{jin}=k)\right.$$
$$\left.+\,\mathbb{E}_{q_\sigma}[\log p(\mathbf{w}_{ji}|\sigma_t)]\right\}. \quad (6.9)$$

Straight forwardly, an image label is assigned according to the image-cluster weight in the group, the segment-cluster counts *and* the likelihood of the image observations. This is quite a sensible way of combining the GMC and SCM, using just the likelihoods under the different cluster representations. $\mathcal{Z}_{y_{ji}}$ is a normalisation constant,

$$\mathcal{Z}_{y_{ji}}=\sum_{t=1}^{T}\exp\left\{\mathbb{E}_{q_\pi}[\log\pi_{jt}]+\sum_{k=1}^{K}\mathbb{E}_{q_\beta}[\log\beta_{tk}]\sum_{n=1}^{N_{ji}}q(z_{jin}=k)+\mathbb{E}_{q_\sigma}[\log p(\mathbf{w}_{ji}|\sigma_t)]\right\}. \quad (6.10)$$

All of the expectations used in the preceding equations are given in Appendix A. The reader is referred to previous chapters for the rest of the VBE and variational Bayes maximisation (VBM) updates.

A cluster splitting heuristic can be employed to find clusters at both the image and segment level with this model, unlike the SCM. Surprisingly, it was found that the best results were still achieved when the image clusters were randomly initialised at some truncation level, $T_{trunc} \gg T$. They are then naturally pruned by VB. Unfortunately, the deterministic nature of the algorithm is again compromised. The greedy cluster splitting heuristic in Algorithm Algorithm 5.1 is used again here for segment-cluster searching, with the same selection criteria in Equation 5.19.

6.4 Image Representation

The same sparse code spatial pyramid matching (ScSPM), and modified ScSPM image features used in Chapter 3 and Chapter 4 are used here for $\mathbf{w}_{ji}$. Hence, the same full-covariance Gaussian cluster representation is used, with Gaussian-Wishart priors.

These priors are scaled using the primary Eigenvalue of the covariance of the data, and tuned with the parameter $C_{width,i}$. As mentioned previously, this is coupled with a Multinomial cluster representation. The hyper-parameter $\phi = 1$ is not varied in the following experiments for simplicity, and because $C_{width,i}$ seemed to have more influence.

Similarly, the same mean-pooled independent component analysis (ICA) representation is used for image segments/super-pixels, $\mathbf{x}_{jin}$ as in Chapter 5. Also a Gaussian cluster representation with Gaussian-Wishart priors are used. These priors are tuned with the parameter $C_{width,s}$, they are not scaled since principal component analysis (PCA)-whitening is used on the features.

Ideally these feature extraction methods would be combined. For example, the scale-invariant feature transform (SIFT) descriptors in the ScSPM may be replaced with ICA responses, and the spatial pyramid pooling would involve the super-pixels at the lowest level. The MCM could then access these features at different levels. However, to stay consistent with previous chapters, this chapter will use the tried and tested methods, and leave this integrated feature extraction method for future work.

6.5 Experiments

In this section the MCM is compared to the models used in previous chapters;

BGMM [5, 12]. This is used for clustering segments in some experiments, and images in others. No context can be utilised by this model. The greedy cluster splitting heuristic is used.

GMC from Chapter 4. This is used to cluster segments, with images forming the "groups", thereby using image context. It is also used to cluster images in groups/albums, which is its original use in Chapter 4. The way in which it is used is made clear in each experiment. Again the greedy cluster splitting heuristic is used for this model

SCM from Chapter 5. This model is used in the same way as the MCM, but cannot make use of the image observations, $\mathbf{w}_{ji}$. Image cluster, segment cluster co-occurrence, and group context can be used by both of these models.

It would be desirable to compare these models to other supervised models in the literature, such as [39, 72], in similar fashion as Chapter 3 and Chapter 4. Unfortunately there are two main obstacles to this. The first is that many of the segmentation results are presented purely qualitatively or only one object per scene. Secondly, many models are sufficiently unique to a specific dataset, or type of data, to make completely fair comparisons with unsupervised methods non-trivial (or the data is not in a readily accessible form).

The same four datasets used in the last chapter will be used for comparisons; (1) the same subset of the Microsoft Research Classes v2 (MSRC-2) used in [39, 69]. (2) The outdoor scenes dataset [84], with segment labels from LabelME [96]. (3) A scientific dataset comprising images taken from multiple autonomous underwater vehicle (AUV) surveys of deep photic zone reefs off of the East coast of Tasmania. (4) the photo albums dataset from Chapter 4. Unfortunately, like the SCM there is not a closed form solution for MCM log-likelihood, so the photo-albums experiment is purely qualitative.

Normalised mutual information (NMI)/V-measure [92, 105] has again been used to validate image and segment clusters against human-labelled ground truth where available. 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 order to assign a segment a ground-truth label, the mode of the pixels in the segment had to be of that label type.

For all experiments, the MCM and SCM were run from 10 random initialisations of the image cluster indicator parameters, $\mathbf{Y}$, with an image cluster truncation level of $T_{trunc} = 100$ for the SCM and $T_{trunc} = 30$ for the MCM. This lower value was chosen for the MCM because full covariance Gaussian-Wishart priors had to be updated in addition to Dirichlet. The GMC and BGMM are both entirely deterministic, with

run times that barely varied, and so were only run once for experimental evaluation.

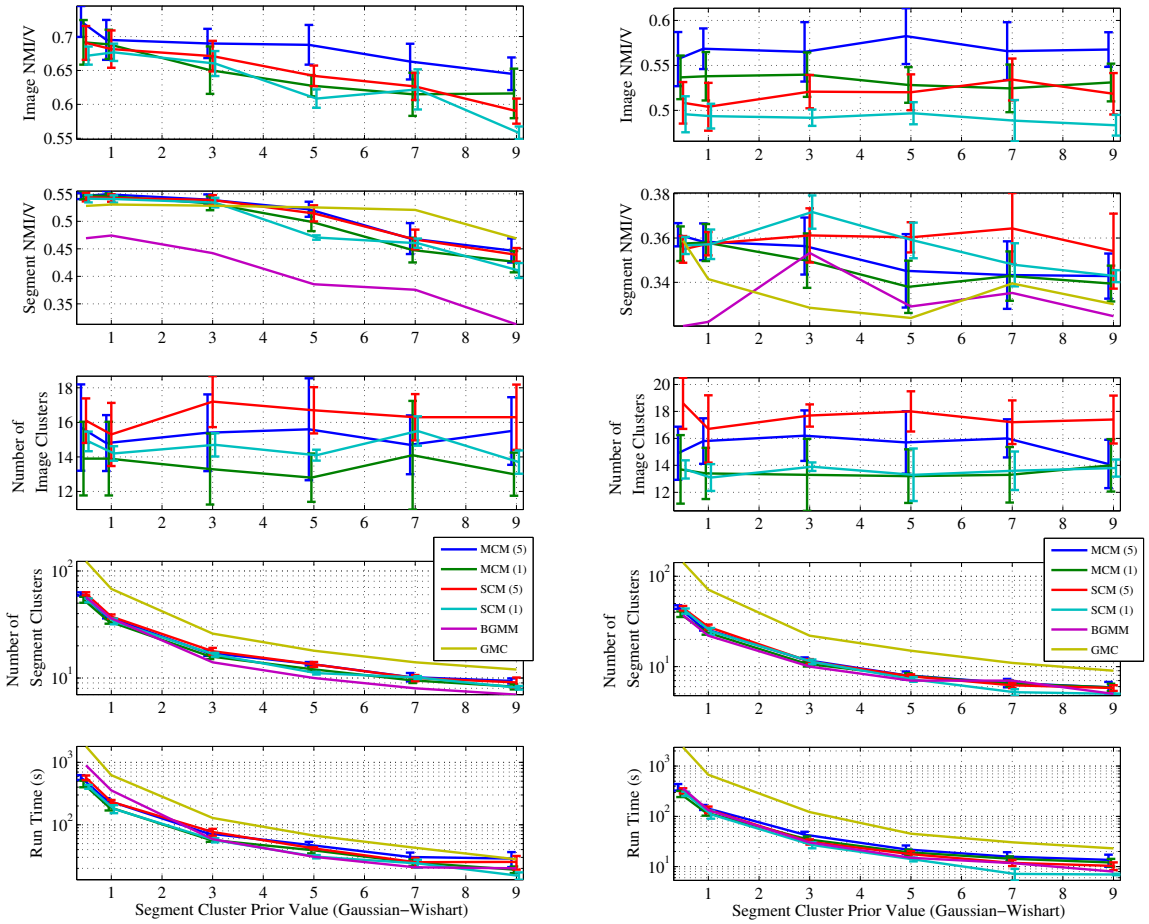
All of the probabilistic models tested are implemented in multi-threaded C++ code, and share as much code as possible. The manner in which all of the algorithms are parelellised is different, so in the interest of fairness, only one thread is used in these experiments for runtime comparison (unless otherwise stated). The MSRC-2, outdoor scenes, and photos albums datasets were run on a 2.8 GHz Intel Core 2 Duo processor, and the AUV dataset on a 3.0 GHz Core 2 Duo.

6.5.1 Effects of Clustering Observations of Images and Image Parts

In this section all models are compared on the same MSRC-2 and outdoor scenes subsets from Chapter 5. Exactly the same segment/super-pixel descriptors are used in both cases. Again PCA whitening is used, with $D_2 = 15$ dimensions preserved. Also the original ScSPM framework from [128] is used for image descriptors, with the same settings used in Chapter 4. These descriptors are compressed with PCA to $D_1 = 20$.

The first experiment performed on these datasets is a repeat of that in Section 5.6.1, in which each of the models is used to cluster the image segments observations, $\mathbf{x}_{jin}$. The SCM can simultaneously find image clusters from only the segment cluster proportions present within each image. The MCM can also use the image observations $\mathbf{w}_{ji}$. Neither of these datasets have natural albums or groups, and so following Chapter 4, the datasets were randomly split into 5 groups, with only a random subset of the true image classes in each group. This random subset division gave best results in Chapter 4. The results are summarised in Figure 6.3 for varying segment cluster prior width, $C_{width,s}$.

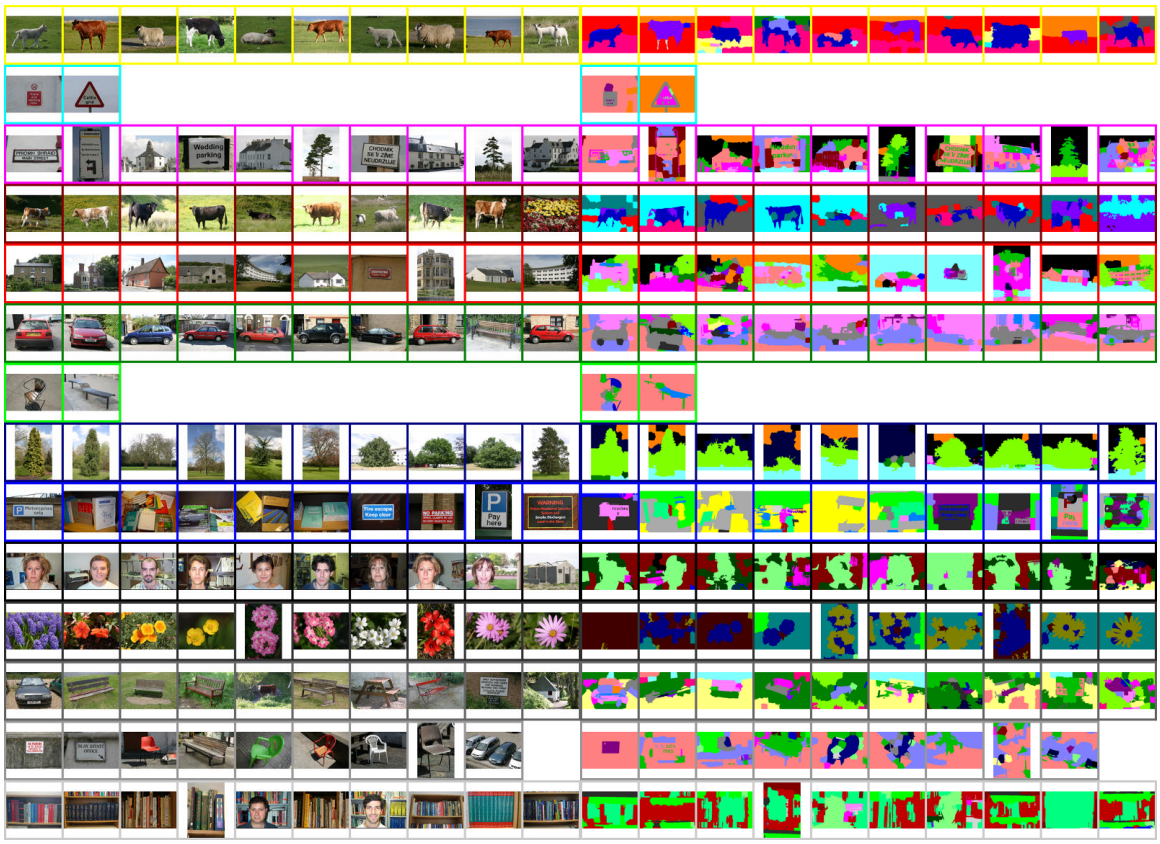
For the MSRC-2 dataset, the SCM and MCM (1) performed very similarly for both clustering images and segments. This is not very surprising since the SCM performed well on this dataset in the last chapter. However, the MCM (5) does seem to be able



(a) MSRC-2 Dataset **(b)** Outdoor Scenes Dataset

Figure 6.3 – Results of modelling different context for clustering images and image segments. MCM (5) and SCM (5) use 5 artificial groups, MCM (1) and SCM (1) use only one group (the original dataset). The GMC and BGMM cannot cluster images, and the BGMM has no notion of separate images. $C_{width,i}$ is 0.06 for MSRC-2, and 0.05 for the outdoor scenes.

to use the context inherent in the groups to improve the image clustering results. The MCM (1) and (5) tend to outperform the SCM (1) and (5) for image clustering in the outdoor datasets. This is not the case for segment clustering performance, which either seems fairly unchanged between the SCM and MCM for the MSRC-2 dataset, and slightly worse for MCM on the outdoor scenes dataset. Interestingly, in both



(a) Sample image clusters **(b)** Corresponding segment clusters

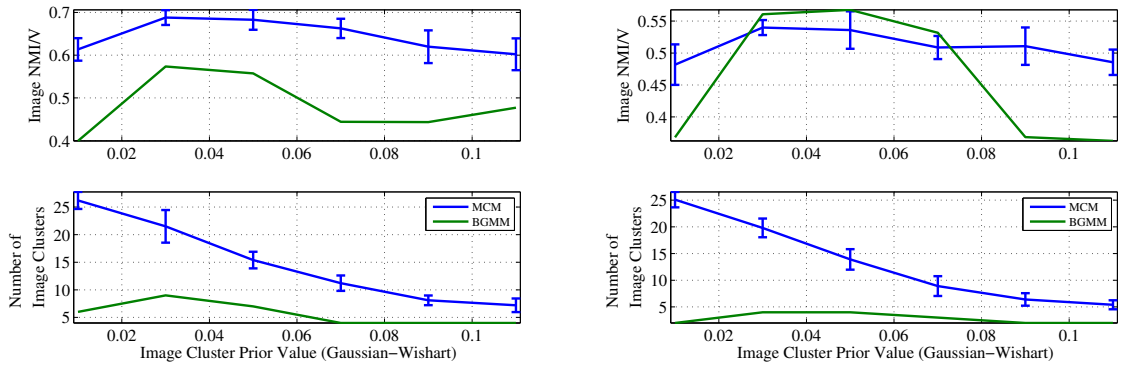
Figure 6.4 – A sample MCM (5) result on the MSRC-2 dataset, with $C_{width,i} = 0.06$, $C_{width,s} = 1$ and $\text{NMI}_i = 0.720$, $\text{NMI}_s = 0.551$, $K = 36$, and $T = 14$. Random image cluster samples are shown in (a) across the rows, and the corresponding segment clusters are shown in (b).

datasets, the SCM and MCM seem to have very similar run-times. It was expected that the MCM would have longer runtime because of the added complexity. A sample MCM clustering result is presented in Figure 6.4 and Figure 6.5.

The second experiment concentrates on comparing the BGMM and the MCM for image clustering performance. So in this experiment the BGMM clusters $\mathbf{w}_{ji}$. No artificial groups were used in this experiment for brevity (this is left to the next experiment with real groups). Results are summarised in Figure 6.6 for varying image cluster width prior $C_{width,i}$.

The MCM maintains a fairly constant NMI in both datasets when compared to the

Figure 6.5 – The segment clusters corresponding to Figure 6.4 shown independently of the image clusters. The rows are random samples of images with a segment cluster present within them. Only the 12 most frequent segment clusters are shown.



(a) MSRC-2 Dataset **(b)** Outdoor Scenes Dataset

Figure 6.6 – Comparing the MCM and BGMM for image clustering. Here the image descriptors are used with the BGMM. $C_{width,s} = 0.5$ was used on the MSRC-2 dataset, and $C_{width,s} = 1$ was used on the outdoor scenes dataset for the MCM.

BGMM, despite the low number of images (302 in MSRC-2 and 320 in the outdoor scenes). It also consistently finds more clusters for the corresponding values of $C_{width,i}$.

6.5.2 Case Study on a Scientific Dataset

For this experiment the same five-dive dataset from Section 5.6.4 is used. Modified ScSPM features were extracted also from the partially corrected colour images, like the ICA descriptors, and the same settings for the ScSPM were used as in all other experiments.

Like in the previous section, the first experiment is simply a repeat of that in Section 5.6.4, with the addition of the MCM. The results are summarised in Figure 6.7 for varying $C_{width,s}$. We can see a significant improvement in results over the SCM for image clustering performance. Taking into account groups, the MCM (5) also tends to sometimes improve results over the MCM (1). This is a very welcome result, since the SCM seems to show the opposite behaviour on this dataset. Segment clustering results are mostly on-par with the SCM – despite the improved image clustering performance. Perhaps the SCM provides sufficient image cluster and object co-occurrence contextual information, so better image clusters do not further impact segment clustering results.

What is also pleasing about these results is the reduced number of image clusters found by the MCM, with better NMI, compared to the SCM. As in the previous section, runtime is very comparable to the SCM. A sample MCM clustering result is presented in Figure 6.8 and Figure 6.9. Also presented in Figure 6.10 is the proportions of the image and segment clusters within each of the AUV dives. We can see that there is quite a bit of cluster sharing between dives, and that not many clusters are specific to a single dive.

The second experiment in this section is similar to the image clustering experiment in the previous section, but this time the BGMM, GMC, MCM (1) and (5) are compared for clustering images for varying $C_{width,i}$. Again, the BGMM and GMC just cluster $\mathbf{w}_{ji}$. Results are summarised in Figure 6.11. The performance of these models is very

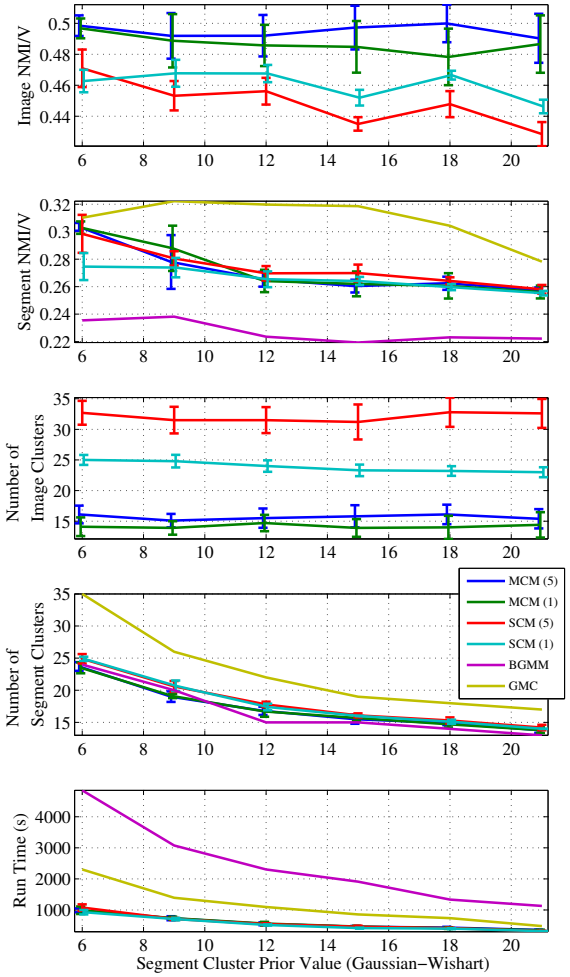
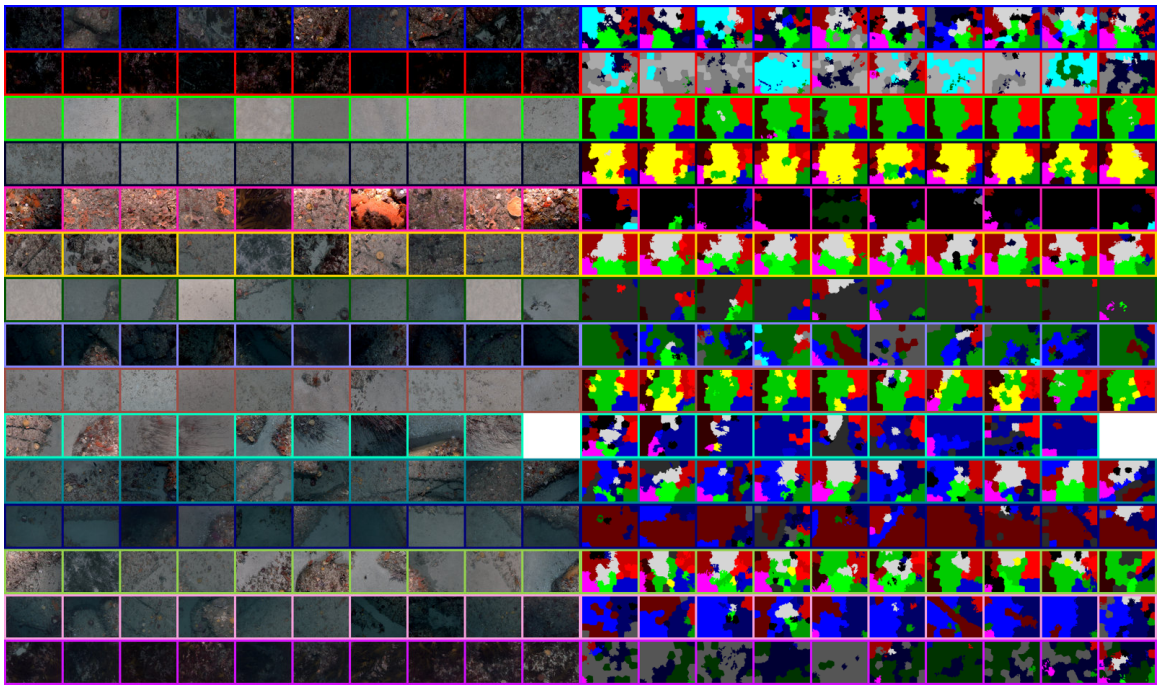


Figure 6.7 – The MCM on the Tasmania AUV dataset. A $C_{width,i} = 0.05$ was used. The models with (5) use the real AUV dives as groups, those with (1) or no number use a concatenation of the whole dataset.

similar for most values of $C_{width,i}$. The models that can take into account groups may slightly outperform those that don't, but not by a large margin. However, the value of $C_{width,s}$ chosen for this experiment (6) also showed the least amount of NMI separation between the MCM (1) and MCM (5) in Figure 6.7. Like the corresponding experiment in the last section, the MCM finds more image clusters for a given value of the prior than the more simple BGMM and GMC, which only use $\mathbf{w}_{ji}$.



(a) Sample image clusters **(b)** Corresponding segment clusters

Figure 6.8 – MCM sample clustering results on the AUV dataset. Random images from the image clusters are shown in (a), and the corresponding segment clusters in (b). These are from the MCM (5) with $C_{width,i} = 0.05$, $C_{width,s} = 6$, $\text{NMI}_i = 0.506$, $\text{NMI}_s = 0.304$, $K = 24$ and $T = 15$.

In this experiment the MCM has demonstrated again that it is a more practically useful model than the SCM, especially in regards to clustering images. It is also keeps pace with the GMC and BGMM in this experiment for clustering images, though does find more clusters than both of these models. Interestingly, the GMC still outperforms the MCM for segment clustering performance on this dataset[2]. Since there are approximately 50 super-pixels per image to cluster as well as each image, the SCM and MCM do take much longer than the models that just cluster image observations. For example, the SCM and MCM take almost 1000 seconds per run on this dataset with these values for $C_{width,\cdot}$ (see Figure 6.11), whereas the GMC and BGMM only take a few seconds. However, a more comprehensive image summary is achieved by the MCM than either the BGMM or the GMC since objects within

[2] As mentioned previously, better segment clustering performance is achieved all-round with mean-shift segments [32].

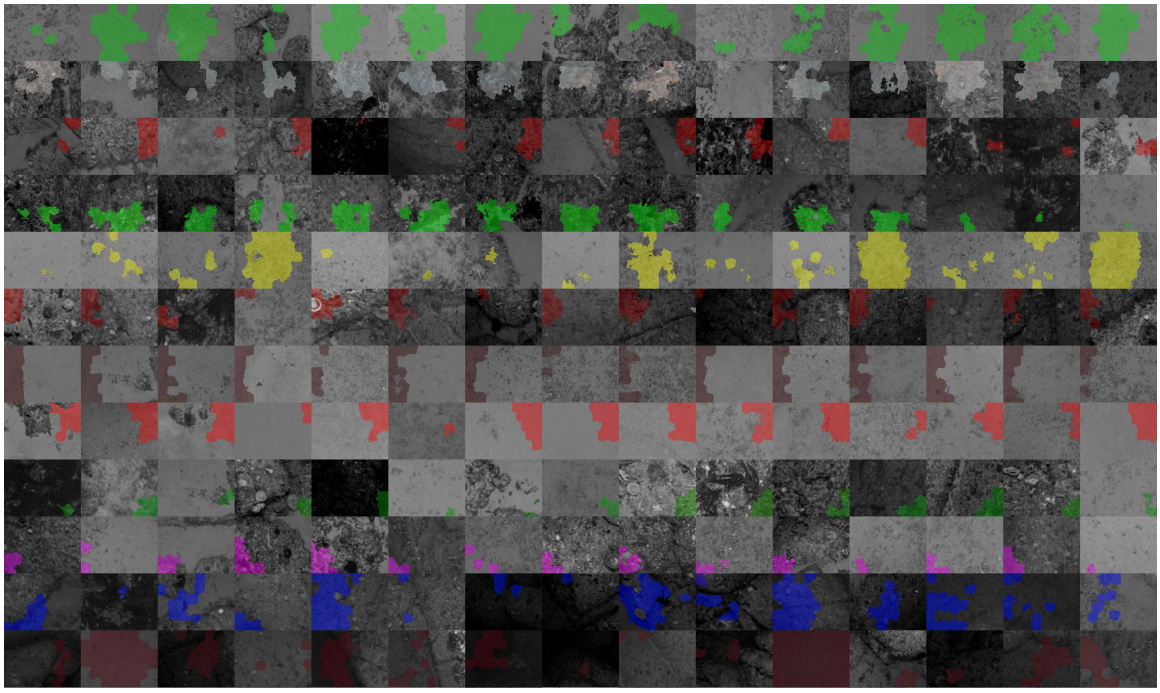
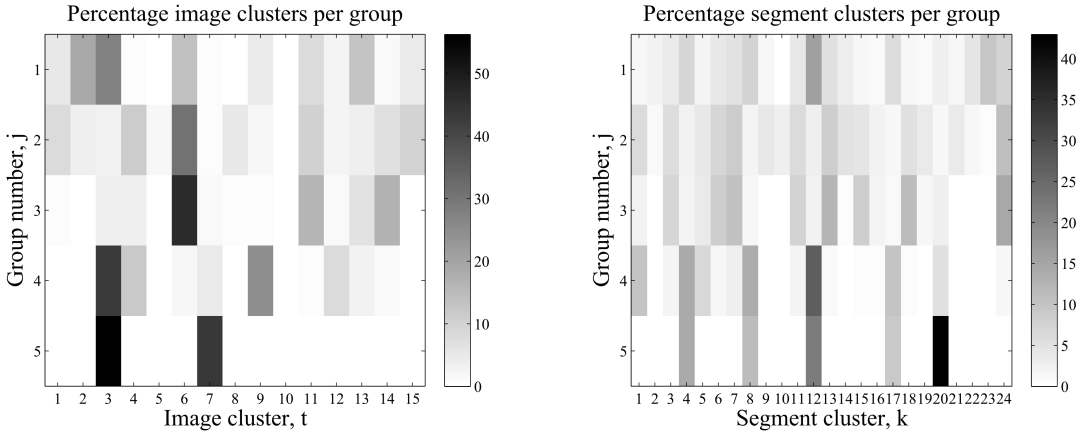


Figure 6.9 – The segment clusters corresponding to Figure 6.8 shown independently of the image clusters. The camera distortion influence on the segment clusters is fairly apparent in some of these clusters. The 12 most frequent segment clusters are shown.



(a) Image clusters per dive **(b)** Segment clusters per dive

Figure 6.10 – Image and segment cluster distributions per group or AUV dive for the MCM result shown in the previous figures.

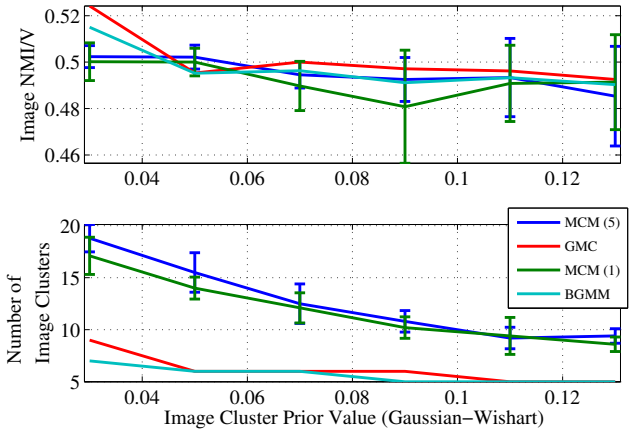


Figure 6.11 – Comparing the MCM, GMC and BGMM for image clustering on the AUV dataset. Here the image descriptors are used with the GMC and BGMM. $C_{width,s} = 6$ was used for the MCM.

images are also discovered and modelled.

6.5.3 Case Study on a Photo Collection

The same 12 holiday photo albums dataset from Section 4.6.4 and Section 5.6.5 was used again here with the MCM. Again, because of the interaction between $\mathbf{Y}$ and $\mathbf{Z}$, there is no closed form solution for the likelihood in the MCM. So, this experiment is also purely qualitative in nature.

In addition to the ICA segment descriptors used by the SCM, the same modified ScSPM descriptors were used from Section 4.6.4 for $\mathbf{w}_{ji}$. The segment features take a little less time to extract than the ScSPM features, and both take around 1 second per image.

Multi-threading was also used in this example, and kept two cores almost constantly under 100% load, like the SCM. A sample MCM result is shown in Figure 6.12, which used $C_{width,i} = 0.15$ and $C_{width,s} = 20$. This took 2465 seconds (41 min, 5 sec) and found $T = 27$ image clusters, and $K = 23$ segment clusters. Also shown are the top seven tags by frequency in each of the image clusters.

Many of the image clusters found by the MCM are similar to the GMC and variational

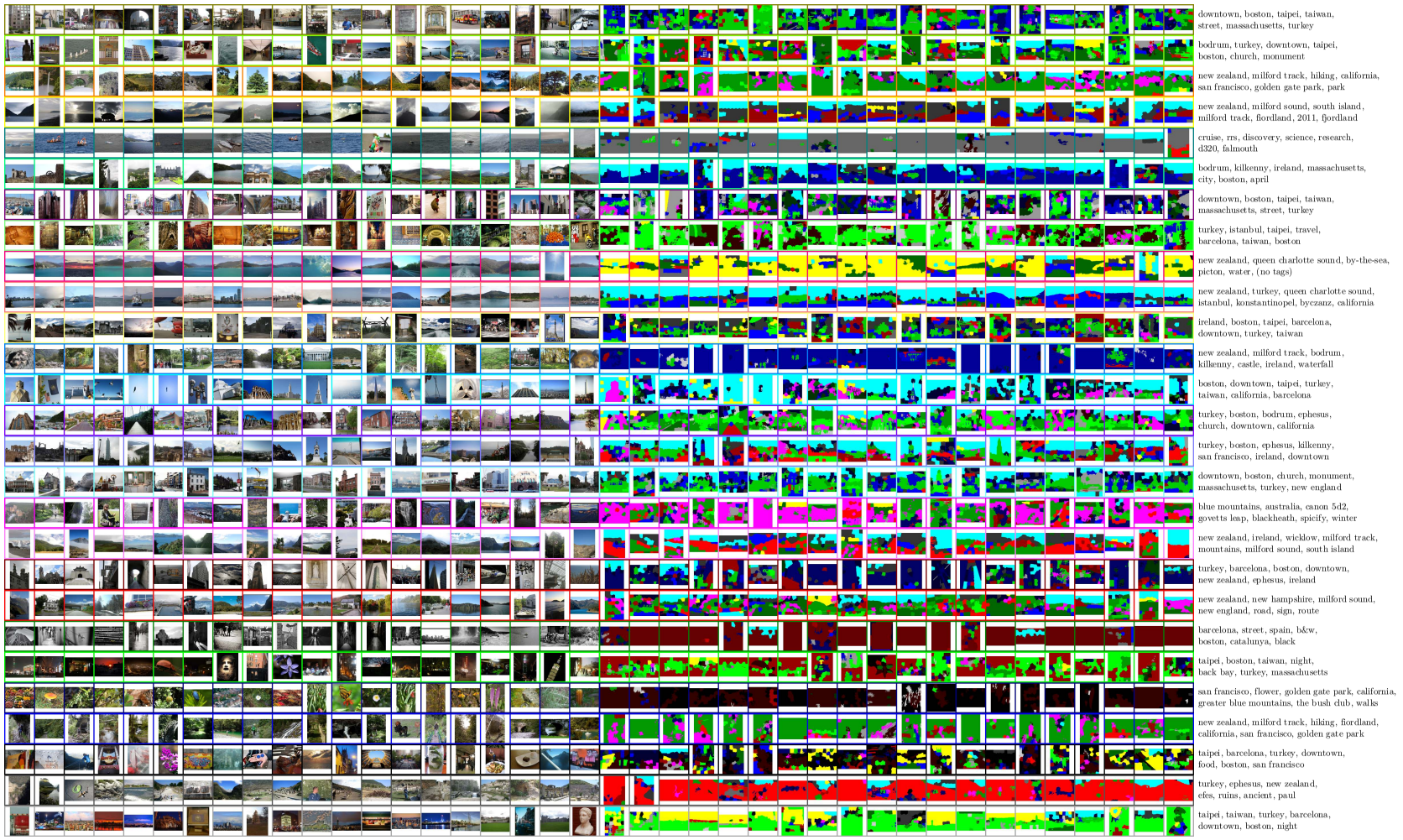


Figure 6.12 – Sample MCM clustering results on the photo album dataset. Random images from each cluster are shown, with the corresponding segment clusters. The seven most frequent tags per image cluster are also shown. There are 27 image clusters, and 23 segment clusters, see text for more details.

Dirichlet process (VDP) in Section 4.6.4, which is unsurprising since the same image descriptors are used. However, there are a few that are fairly different, for instance the MCM finds clusters that may be more colour based, such as the black-and-white image cluster in Figure 6.12 (7th from the bottom). This is similar to the type of image cluster found by the SCM in Section 5.6.5. It is also appears more successful at finding clusters which have a single small foreground object, such as flowers (5th from bottom) as their focus. These two clusters were found fairly consistently between multiple runs, and prior settings. Overall, the image cluster found by the MCM are more convincing than those of the SCM, and appear no worse than those found by the GMC and VDP. This can largely be attributed to the ScSPM descriptor's ability to model crude image spatial layout.

6.6 Summary

By combining the GMC from Chapter 4 and the SCM from Chapter 5, the MCM was created. It can be seen that by simultaneously clustering whole image and segment observations the advantages of both the GMC and SCM are present in the MCM. A richer image representation is retained, since images can be described as combinations of "objects", while at the same time the number of image clusters can be effectively controlled while retaining more consistent performance. Additionally, the MCM can now take advantage of the context inherent within groups or albums for improving clustering results. It was not apparent from the last chapter that the SCM could do the same.

The dual Multinomial-Gaussian representation of images of the MCM was often similar in performance to that of the GMC and BGMM (using ScSPM descriptors), if occasionally a little worse. However, in many cases the MCM was more consistent in its performance than the other models with respect to the setting of $C_{width,i}$. This was especially evident when the number of images in a dataset is small. In these cases the evidence from the more numerous super-pixel or segment observations could be used to improve the image clustering result. Unfortunately, the enhanced image clustering

performance did not seem to improve segment clustering results over the SCM. Perhaps the SCM image representation was sufficient in capturing the scene type, and object co-occurrence contextual information needed to improve segment clustering[3].

Interestingly, there did not seem to be much difference in the run-time between the MCM and SCM for the smaller experiments, despite the MCM having an extra full-covariance Gaussian-Wishart distribution to update per image cluster. The MCM did usually find less image clusters than the SCM, and its truncation level was set lower, which could account for the runtime similarity. It is expected this runtime would diverge for larger datasets though, like the photo albums dataset. In general, the MCM is a more practical alternative to the SCM, as long as the extra time to calculate the image features can be afforded.

As stated previously, future work could include combining the image ScSPM descriptor with the segment pooled ICA descriptors. The MCM could then utilise separate layers of this new descriptor representation, while also reducing the computational time in producing these descriptors.

Unsupervised image "understanding" algorithms can find sensible representations of images, even in absence of any semantic knowledge of a scene or its constituent parts. As future work it would be useful to extend these algorithms to leverage annotation data where available. Their strong modelling capability of the visual aspects of image datasets may be used to make them robust to incorrect or noisy annotation data, like in [72], but now at *both* object and scene levels.

[3]In results generated subsequent to this thesis using mean-shift segments [32] and larger datasets, the MCM does convincingly outperform the SCM for segment clustering.

Chapter 7

Conclusion

This thesis has been concerned with exploring, evaluating, and designing algorithms for unsupervised modelling of visual data. Works such as [117] show that this is a very challenging problem, as has been found in this thesis. Interestingly, this is also a relatively unexplored problem in the vision literature compared to supervised and semi-supervised learning. The contributions of this thesis in this regard will hopefully help practitioners and researchers to understand and better tackle complex, high level, fully unsupervised vision modelling problems.

This chapter concludes the thesis. A meta-summary of the contributions and findings of Chapters 3-6 is given in Section 7.1, and potential future work following on from this thesis is summarised in Section 7.2.

7.1 Summary of Contributions

The contributions of this thesis arise from applying, evaluating and creating machine learning algorithms to unsupervised modelling of visual data. The following is a summary of the principal contributions in this thesis.

7.1.1 Large Scale Adaptation and Analysis of Sparse Coding Spatial Pyramids

The popular sparse code spatial pyramid matching (ScSPM) framework introduced by [128] was adapted such that it could be used for large scale classification and clustering tasks. In its original implementation, it is computationally slow to learn a dictionary and encode image patches, limiting its use for truly large datasets. Furthermore, it creates very high dimensional descriptors (on the order of 20,000 dimensions as used in this thesis), which makes it impractical for most clustering algorithms.

In Chapter 3 it was shown that the image descriptors created by the ScSPM framework are very amenable to compression with fast linear dimensionality reduction techniques such as (iterative and probabilistic) principal component analysis (PCA). For classification tasks, the same performance can be achieved as using the original codes, with only a fraction of the dimensionality. However, it was also found that the compressibility was a function of the number of classes in the dataset. This compressibility admitted the use of these highly effective image descriptors for clustering applications.

It was shown that the sparse coding (SC) patch encoding algorithm could be replaced with the faster, more scalable orthogonal matching pursuit (OMP) in this framework. There is a little reduction in performance for classification, but almost no discernible difference for PCA-reduction and K-means clustering. Furthermore, the dictionary used for encoding scale-invariant feature transform (SIFT) patches to sparse codes did not have to originate from the same dataset as these patches. There was little evidence for classification performance loss in this scenario, as long as the dataset used to obtain the dictionary was diverse in appearance, which is readily quantifiable. In fact, there was more evidence found for the choice of dictionary learning algorithm impacting performance for classification and clustering, somewhat contrary to [33].

The contribution of this chapter is of an empirical nature. That is, a thorough empirical evaluation of these frameworks for potentially large scale applications was presented. This is useful knowledge for practical use of these frameworks, and may

also lend itself to incremental classification and clustering applications, since the dictionary does not necessarily have to be relearned.

7.1.2 Clustering Multiple Related Datasets Jointly

It was hypothesised that jointly clustering multiple related visual datasets, i.e. datasets that exhibit similar images, while keeping the proportion of the clusters in each dataset unique, could simplify the discovery of these clusters. The intuition behind this being that these datasets, if sufficiently diverse, could provide different views of these clusters of observations in feature space. This could be especially helpful in the case of highly overlapping clusters in feature space, which may not co-occur in particular datasets. In Chapter 4 this hypothesis was shown to be true, but depended heavily on the composition of the datasets to be jointly clustered. Datasets that exhibited the same proportions of the latent ground truth classes provided no benefit, or added "contextual" information. However, datasets that had different distributions, and even a subset, of the truth classes definitely aided clustering. Even in very large datasets, where the clusters had a lot of observations belonging to them (hence making them easier to find), the models that could take advantage of this structure, i.e. the grouped mixtures clustering model (GMC) and its variants, were computationally faster than conventional clustering algorithms.

It was also noticed in Chapter 4 that when using Bayesian mixture models with Gaussian mixtures and large datasets, the choice of distribution over the mixture weights, $\boldsymbol{\pi}$, did not have a large impact on results. This was especially true for conventional mixture models such as the Bayesian Gaussian mixture model (BGMM) (Dirichlet) and variational Dirichlet process (VDP) (Dirichlet process). This is likely because of the model likelihood, and especially the component of likelihood arising from the Gaussian clusters, overwhelming the influence of the mixture weight priors.

7.1.3 An Analysis of Context and Simultaneous Clustering

The GMC was extended to simultaneously cluster image parts (segments/super-pixels), and images, while retaining the ability to jointly cluster multiple datasets. This model is referred to as the simultaneous clustering model (SCM). In this model the image-parts were modelled as Gaussian clusters, while the images are effectively Multinomial clusters. This made a large difference to some of the previously seen behaviour of the GMC. For instance, unlike the GMC, now the choice of group mixture weight prior (Dirichlet vs. generalised Dirichlet) had a substantial effect on the image clustering results. This may be explained by the Multinomial clusters' likelihood and complexity penalties not dominating the mixture weight priors in the free energy objective function nearly as much as Gaussian clusters. Similar effects have been seen before in the text modelling literature between latent Dirichlet allocation (LDA), which uses a symmetric Dirichlet prior, and the hierarchical Dirichlet process (HDP) which uses a hierarchical Dirichlet process prior. A Dirichlet process is similar in some ways to the generalised Dirichlet. It was also observed that these Multinomial clusters did not seem to benefit as much from clustering in groups as the Gaussian clusters, if indeed at all.

Despite these differences, the GMC was seen to exhibit similar benefits when applied to clustering image-parts in image context as when it was applied to images in album/survey context. The SCM, also showed benefits over a BGMM for clustering image-parts, even though it used image-clusters as the context. In many cases it was competitive with the GMC for this, and if it did do worse, its runtime was always significantly faster. This is probably because it had fewer distributions over image-part cluster weights to update in its maximisation step (the GMC has one per image, the SCM one per image-cluster). It also tended to find fewer image-part clusters. These studies show that context, modelled in multiple and entirely unsupervised ways drastically improves upon conventional clustering. Furthermore, this thesis has shown it is possible to model images at multiple levels in a totally unsupervised manner, which is something not thoroughly explored in the literature previously.

7.1.4 Combining Models for a Richer Image Representation

The SCM and GMC were combined into a unified model referred to as the multiple-source clustering model (MCM). This model could take advantage of whole image descriptors, like the GMC, which used a Gaussian mixture cluster representation. This representation was shown to work well with top-level groups or albums for improved clustering, unlike the SCM. The MCM also retained the ability of the SCM to provide a rich description of image clusters as a combination of constituent image-part (segment) clusters or "objects". While supervised models with similar capabilities have been used previously, to the author's knowledge this is the fist time such a model has been formulated and applied to fully unsupervised problems. The MCM worked consistently and effectively on large, visually unconstrained datasets, unlike the SCM. Furthermore, this model also worked well on datasets with very few images, where the GMC and regular Bayesian mixture models sometimes find very few image clusters.

7.2 Future Work

This section summarises future work, potential contributions, and interesting avenues of research that naturally follow on from the work in this thesis.

7.2.1 Integrating Sparse Coding, Pooling and Dimensionality Reduction

Spatial pyramid pooling is somewhat heuristic. It would be interesting to follow on from the work of Boureau et al. [26, 27] to further understand why this max-pooling spatial pyramid framework yields good performance for classification and clustering. Furthermore, if suitable "generative" forms of this pooling could be found, such as [67], it would be interesting to see if sparse coding, pooling and dimensionality reduction could be combined into one framework, extending the work of Gkioulekas and Zickler [49]. Additionally, it would useful if this framework could be extended

to a multi-layered model like those in [20, 67], so SIFT descriptors do not have to be used at all.

Another option, mentioned in Chapter 6, may be to combine ScSPM with the independent component analysis (ICA) based segment descriptor framework introduced in Chapter 5. The SIFT patch descriptors used by ScSPM may be replaced with dense ICA patch codes, which are also learned from the images, and include colour information. Spatial pyramid pooling with an initial super-pixel or segment layer, rather than grids, may also improve results. Then models such as the MCM can use the direct output of different layers of one hierarchical descriptor framework.

7.2.2 Clarifying the Relationship Between Groups, Classes, and Distributions

In Chapter 4 it was noticed that there was a relationship between the number of groups (albums/surveys), and the performance of the GMC. After the dataset was divided into a certain number of groups, the performance plateaued. It would be interesting to establish whether this plateauing effect is related to the number of latent classes within each dataset, or if the relationship is more complex.

In Chapter 5 and Chapter 6 it was noticed that the SCM did not take advantage of the contextual information inherent within groups of data unlike the GMC and MCM. This is most likely because Multinomial cluster distributions are used exclusively for describing image clusters as opposed to Gaussian in the GMC in Chapter 4, and joint Multinomial-Gaussian in the MCM. Additionally, the image-part, or super-pixel/segment, Gaussian clusters in the SCM and MCM seem to benefit greatly from the image-cluster context. This is similar to the GMC when used for image-part clustering with only image context (images as groups of segments) as in Chapter 5. It would be desirable to clarify exactly why these distributions behave differently with groups; is it a fundamental limitation of the Multinomial representation, or is it linked to Multinomial clustering being influenced differently than Gaussian clustering by the mixture weights?

7.2.3 Exploiting more Context within Images

In the literature there are other ways to exploit context within imagery to enhance classification performance. They have not been used here because they have either been fairly thoroughly investigated in the past, make the models used far more complex, and/or make inference more computationally demanding. However, it may be worthwhile investigating the following types of context in the future for models like the SCM and MCM:

- While the SCM and MCM do model rudimentary object co-occurrence in their image clusters, there are numerous other ways of modelling object correlations. Models like correlated topic model (CTM) [16] or Pachinko allocation model (PAM) [74] can also model "topic" (object) correlation, as a pair wise covariance, or an arbitrary directed acyclic graph (DAG) respectively[1]. Unfortunately each have their own caveats, and trade-offs would need to be made to incorporate them into the SCM. It is expected this will benefit unsupervised modelling, especially in the case of the autonomous underwater vehicle (AUV) data, as many organisms are dependent on the substrate on which they occur.

- An alternate representation of album and image context. Perhaps something like a HDP, which can model image parts as having a "local" context at the image level (like the GMC), and a more "global" context at the group album level. This model is similar to the SCM, with the exception of not simultaneously clustering images, but rather having a mixture model for every image.

- Spatial context (within image), whether it be using a smoothing type approach using random fields [64, 132], or hierarchical priors [4, 39, 122], or placing objects in particular locations in scenes [82, 106, 116]. It is unclear whether these models will benefit the AUV imagery, since it is quite unstructured. Adding in spatial constraints usually has a large impact on computational runtime. Although, the MCM does make use of the ScSPM image descriptor, which does include a crude notion of image spatial layout.

[1] Also see the tree-like model in presented by Choi et al. [31]

7.2.4 Extensions to Semi-Supervised Learning

One of the most useful extensions to this work would be enabling the models presented to take advantage of image annotations when they exist. A potential contribution here could be to handle "noisy" labels (incorrect labels) at multiple levels in an image, such as scene level and object level. There is quite a bit of literature dealing with noisy labels, most notably [72], however this is usually at a single image level.

This would be especially useful for the AUV datasets, where only a small fraction of organisms in some of the images are labelled, and the image/substrate labels have a high proportion of error. An active learning approach could be adopted here too, where these models could alert an expert that certain labels may be incorrect. These sorts of models could potentially also be used to adaptively collapse the (usually large) taxonomy tree to the most likely species, in effect suggesting labels, for the image parts currently being labelled by humans.

Bibliography

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. SLIC superpixels compared to state-of-the-art superpixel methods. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(11):2274 –2282, Nov. 2012. ISSN 0162-8828.

[2] M. Aharon, M. Elad, and A. Bruckstein. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *Signal Processing, IEEE Transactions on*, 54(11):4311–4322, 2006.

[3] H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723, 1974.

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

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

[6] F. Bach. Sparse methods for machine learning theory and algorithms (tutorial). In *Advances in Neural Information Processing Systems*, December 2009.

[7] R. G. Baraniuk and M. B. Wakin. Random projections of smooth manifolds. *Foundations of Computational Mathematics*, 9(1):51–77, 2009.

[8] R. G. Baraniuk, M. Davenport, R. DeVore, and M. Wakin. A simple proof of the restricted isometry property for random matrices. *Constructive Approximation*, 28(3):253–263, 2008.

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

[10] M. Belkin and P. Niyogi. Laplacian Eigenmaps and spectral techniques for embedding and clustering. In *Advances in Neural Information Processing Systems*, volume 14, pages 585–591, 2002.

[11] Y. Bengio. Learning deep architectures for AI. *Foundations and Trends in Machine Learning*, 2(1):1–127, 2009. ISSN 1935-8237.

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

[13] D. M. Blei. *Probabilistic models of text and images*. PhD thesis, University of California, Berkeley, 2004.

[14] 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. ISBN 1-58113-646-3.

[15] D. M. Blei and M. I. Jordan. Variational methods for the Dirichlet process. In *Proceedings of the twenty-first International Conference on Machine Learning*. ACM New York, NY, USA, 2004.

[16] D. M. Blei and J. D. Lafferty. A correlated topic model of science. *The Annals of Applied Statistics*, 1(1):17–35, June 2007. ISSN 1932-6157.

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

[18] L. Bo, X. Ren, and D. Fox. Kernel descriptors for visual recognition. *Advances in Neural Information Processing Systems*, 7, 2010.

[19] L. Bo, K. Lai, X. Ren, and D. Fox. Object recognition with hierarchical kernel descriptors. In *Computer Vision and Pattern Recognition. CVPR. IEEE Conference on*, pages 1729–1736. IEEE, 2011.

[20] L. Bo, X. Ren, and D. Fox. Hierarchical matching pursuit for image classification: Architecture and fast algorithms. In *Advances in Neural Information Processing Systems*, 2011.

[21] A. Bosch, X. Munoz, R. Marti, X. Muñoz, and R. Martí. Which is the best way to organize/classify images by content? *Image and vision computing*, 25 (6):778–791, June 2007. ISSN 02628856.

[22] N. Bouguila. Clustering of count data using generalized dirichlet multinomial distributions. *IEEE Transactions on Knowledge and Data Engineering*, 20(4): 462–474, 2008.

[23] N. Bouguila and D. Ziou. A hybrid SEM algorithm for high-dimensional unsupervised learning using a finite generalized Dirichlet mixture. *IEEE Transactions on Image Processing*, 15(9):2657–2668, Sept. 2006. ISSN 1057-7149.

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

[25] Y.-L. Boureau, F. Bach, Y. LeCun, and J. Ponce. Learning mid-level features for recognition. In *Computer Vision and Pattern Recognition. CVPR. IEEE Conference on*, pages 2559–2566. IEEE, 2010.

[26] Y.-L. Boureau, J. Ponce, and Y. LeCun. A theoretical analysis of feature pooling in visual recognition. In *International Conference on Machine Learning*, pages 111–118, 2010.

[27] Y.-L. Boureau, N. Le Roux, F. Bach, J. Ponce, and Y. Lecun. Ask the locals: multi-way local pooling for image recognition. In *Computer Vision. ICCV. The 13th International Conference on*, Barcelone, Espagne, 2011.

[28] T. Bridge, A. Scott, and D. M. Steinberg. Abundance and diversity of anemonefishes and their host sea anemones at two mesophotic sites on the Great Barrier Reef, Australia. *Coral Reefs*, 31:1057–1062, 2012. ISSN 0722-4028.

[29] W. Buntine. Variational extensions to EM and multinomial PCA. *Machine Learning: ECML 2002*, pages 23–34, 2002.

[30] L. Cao and 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*, pages 1–8, Oct. 2007.

[31] M. J. Choi, J. Lim, A. Torralba, and A. Willsky. Exploiting hierarchical context on a large database of object categories. In *Computer Vision and Pattern Recognition. CVPR. IEEE Conference on*, pages 129–136, June 2010.

[32] 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.

[33] A. Coates and 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, pages 921–928, New York, NY, USA, June 2011. ACM. ISBN 978-1-4503-0619-5.

[34] A. Coates, H. Lee, and A. Y. Ng. An analysis of single-layer networks in unsupervised feature learning. In *Artificial Intelligence and Statistics (AISTATS)*, volume 1001, page 48109, 2011.

[35] P. J. Collins. *Differential and Integral Equations*. Oxford University Press, Great Clarendon Street, Oxford OX2 6DP, 2006.

[36] 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.

[37] T. F. Cox and M. A. A. Cox. *Multidimensional Scaling*. Chapman and Hall, second edition, 2000.

[38] M. A. Davenport, M. F. Duarte, Y. C. Eldar, and G. Kutyniok. Introduction to compressed sensing. *Compressed Sensing: Theory and Applications*, 93, 2011.

[39] L. Du, L. Ren, D. Dunson, and L. Carin. A Bayesian model for simultaneous image clustering, annotation and object segmentation. In *Advances in Neural Information Processing Systems*, volume 22, pages 486–494, 2009.

[40] L. Fei-Fei and 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, and G. Farinella, editors, *Computer Vision*, volume 285 of *Studies in Computational Intelligence*, pages 157–171. Springer Berlin Heidelberg, 2010. ISBN 978-3-642-12847-9.

[41] L. Fei-Fei and P. Perona. A Bayesian hierarchical model for learning natural scene categories. In *Computer Vision and Pattern Recognition. CVPR. IEEE ComputerSociety Conference on*, volume 2, pages 524–531. IEEE, 2005. ISBN 0769523722.

[42] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories. *Computer Vision and Image Understanding*, 106(1):59–70, Apr. 2007. ISSN 1077-3142.

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

[44] B. J. Frey and D. Dueck. Clustering by passing messages between data points. *Science*, 315(5814):972, 2007.

[45] A. Friedman, O. Pizarro, and S. B. Williams. Rugosity, slope and aspect from bathymetric stereo image reconstructions. In *OCEANS*, Sydney, May 2010. IEEE Oceanic Engineering Society.

[46] J. Gao, Q. Shi, and T. S. Caetano. Dimensionality reduction via compressive sensing. *Pattern Recognition Letters*, 2012.

[47] P. Gehler and S. Nowozin. On feature combination for multiclass object classification. In *Computer Vision. ICCV. IEEE 12th International Conference on*, pages 221–228, October 2009.

[48] Y. Girdhar, P. Giguere, and G. Dudek. Autonomous adaptive underwater exploration using online topic modelling. In *International Symposium on Experimental Robotics*, 2012.

[49] I. Gkioulekas and T. Zickler. Dimensionality reduction using the sparse linear model. *Advances in Neural Information Processing*, 2011.

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

[51] K. Grauman and T. Darrell. The pyramid match kernel: discriminative classification with sets of image features. In *Computer Vision. ICCV. Tenth IEEE International Conference on*, volume 2, pages 1458–1465, oct. 2005.

[52] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. Technical Report 7694, California Institute of Technology, 2007. URL `http://authors.library.caltech.edu/7694`.

[53] Q. Gu and J. Zhou. Learning the shared subspace for multi-task clustering and transductive transfer classification. In *Data Mining. ICDM. Ninth IEEE International Conference on*, number 1, pages 159–168. IEEE, Dec. 2009. ISBN 978-1-4244-5242-2.

[54] N. Halko, P.-G. Martinsson, and J. A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. Technical Report arXiv:0909.4061v2, http://arxiv.org/abs/0909.4061, 2009.

[55] X. He and P. Niyogi. Locality preserving projections. In *Advances in Neural Information Processing Systems*, volume 16, pages 153–160, 2003.

[56] T. Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the Twenty-Second Annual International SIGIR Conference on Research and Development in Information Retrieval*, 1999.

[57] L. Hubert and P. Arabie. Comparing partitions. *Journal of Classification*, 2: 193–218, 1985. ISSN 0176-4268.

[58] A. Hyvärinen and E. Oja. Independent component analysis: algorithms and applications. *Neural Networks*, 13(4–5):411–430, 2000. ISSN 0893-6080.

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

[60] K. Jarrett, K. Kavukcuoglu, M. A. Ranzato, and Y. LeCun. What is the best multi-stage architecture for object recognition? In *Computer Vision. ICCV. IEEE 12th International Conference on*, pages 2146–2153. IEEE, 2009.

[61] W. B. Johnson and J. Lindenstrauss. Extensions of Lipschitz mappings into a hilbert space. *Contemporary mathematics*, 26(189-206):1, 1984.

[62] M. Johnson-Roberson, O. Pizarro, S. B. Williams, and I. Mahon. Generation and visualization of large-scale three-dimensional reconstructions from underwater robotic surveys. *Journal of Field Robotics*, 27(1):21–51, 2009.

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

[64] D. Larlus, J. Verbeek, and F. Jurie. Category level object segmentation by combining bag-of-words models with Dirichlet processes and random fields. *International Journal of Computer Vision*, 88(2):238–253, 2010.

[65] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Computer Vision and Pattern Recognition. CVPR. IEEE Computer Society Conference on*, volume 2, pages 2169–2178, 2006.

[66] H. Lee, A. Battle, R. Raina, and A. Y. Ng. Efficient sparse coding algorithms. *Advances in neural information processing systems*, 19:801, 2007.

[67] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 609–616. ACM, 2009.

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

[69] L. Li, M. Zhou, G. Sapiro, and L. Carin. On the integration of topic modeling and dictionary learning. *International Conference on Machine Learning*, 2011.

[70] L.-J. Li and L. Fei-Fei. What, where and who? classifying events by scene and object recognition. In *Computer Vision. ICCV. IEEE 11th International Conference on*, pages 1–8, 2007.

[71] L.-J. Li and L. Fei-Fei. Optimol: automatic online picture collection via incremental model learning. *International Journal of Computer Vision*, 88(2): 147–168, 2010.

[72] 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. IEEE Conference on*, pages 2036–2043, June 2009.

[73] L.-J. Li, C. Wang, Y. Lim, D. Blei, and L. Fei-Fei. Building and using a semantivisual image hierarchy. In *Computer Vision and Pattern Recognition. CVPR. 2010 IEEE Conference on*, pages 3336–3343, June 2010.

[74] W. Li and A. McCallum. *Pachinko allocation: DAG-structured mixture models of topic correlations*. PhD thesis, University of Massachusetts Amherst, 2006.

[75] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[76] D. J. C. Mackay. *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, version 7.2 edition, 2003. ISBN 0521642981.

[77] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297, 1967.

[78] T. J. Malisiewicz, J. C. Huang, and A. A. Efros. Detecting objects via multiple segmentations and latent topic models. Technical report, Massachusetts Institute of Technology, 2006.

[79] T. Masada, S. Kiyasu, and S. Miyahara. Clustering images with multinomial mixture models. In *Proceedings Of The 8th Symposium On Advanced Intelligent Systems (ISIS)*, pages 343–348, 2007.

[80] R. M. Neal and G. E. Hinton. A new view of the EM algorithm that justifies incremental and other variants. In *Learning in Graphical Models*, pages 355–368. Kluwer Academic Publishers, 1993.

[81] A. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems*, pages 849–856, 2001.

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

[83] T. Ojala, M. Pietikäinen, and T. Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987, July 2002.

[84] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42(3):145–175, 2001.

[85] 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.

[86] J. Paisley and L. Carin. Nonparametric factor analysis with Beta process priors. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 777–784. ACM, 2009.

[87] C. Papadimitriou, P. Raghavan, H. Tamaki, and S. Vempala. Latent semantic indexing: A probabilistic analysis. In *Proceedings of ACM PODS*, 1998.

[88] Y. Pati, R. Rezaiifar, and P. Krishnaprasad. Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition. In *Signals, Systems and Computers. Record of The Twenty-Seventh Asilomar Conference on*, volume 1, pages 40–44, Nov. 1993.

[89] J. Pitman and M. Yor. The two-parameter Poisson-Dirichlet distribution derived from a stable subordinator. *Annals of Probability*, 25:855–900, 1997.

[90] O. Pizarro, S. B. Williams, and J. Colquhoun. Topic-based habitat classification using visual data. In *OCEANS*, pages 1–8, 2009.

[91] W. M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):846–850, 1971.

[92] A. Rosenberg and J. Hirschberg. V-measure: A conditional entropy-based external cluster evaluation measure. In *Conference on Empirical Methods in Natural Language Processing*, 2007.

[93] S. Roweis. EM algorithms for PCA and SPCA. *Advances in Neural Information Processinge Systems (NIPS)*, pages 626–632, 1998.

[94] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323, 2000.

[95] 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. IEEE Conference on*, pages 1605–1614. IEEE, 2006.

[96] 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. ISSN 0920-5691. 10.1007/s11263-007-0090-8.

[97] B. C. Russell, A. Efros, J. Sivic, W. T. Freeman, and A. Zisserman. Segmenting scenes by matching image composites. In *Advances in Neural Information Processing Systems*, 2009.

[98] G. E. Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6 (2):461–464, 1978.

[99] M. W. Seeger. Bayesian inference and optimal design for the sparse linear model. *The Journal of Machine Learning Research*, 9:759–813, 2008.

[100] J. Seiler, A. Friedman, D. M. Steinberg, N. Barrett, A. Williams, and N. J. Holbrook. Image-based continental shelf habitat mapping using novel automated data extraction techniques. *Continental Shelf Research*, 45:87–97, 2012. ISSN 0278-4343.

[101] 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. IEEE Conference on*, pages 1–8. IEEE, 2008.

[102] R. Socher and 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*, pages 966–973, June 2010.

[103] D. M. Steinberg, A. Friedman, O. Pizarro, and S. B. Williams. A Bayesian nonparametric approach to clustering data from underwater robotic surveys. In *International Symposium on Robotics Research*, Flagstaff, AZ, Aug. 2011.

[104] M. Steyvers, P. Smyth, M. Rosen-Zvi, and T. Griffiths. Probabilistic author-topic models for information discovery. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 306–315. ACM, 2004.

[105] A. Strehl and J. Ghosh. Cluster ensembles – a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, 3: 583–617, Mar. 2003. ISSN 1532-4435.

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

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

[108] E. B. Sudderth, A. Torralba, W. Freeman, and A. Willsky. Describing visual scenes using transformed Dirichlet processes. *Advances in Neural Information Processing Systems*, 18:1297, 2006. ISSN 1049-5258.

[109] 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.

[110] Y. W. Teh, K. Kurihara, and M. Welling. Collapsed variational inference for HDP. *Advances in Neural Information Processing Systems*, 20:1481–1488, 2008.

[111] J. B. Tenenbaum, V. De Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500): 2319–2323, 2000.

[112] D. R. Thompson, T. Smith, and D. Wettergreen. Information-optimal selective data return for autonomous rover traverse science and survey. In *Robotics and Automation. ICRA. IEEE International Conferenceon*, pages 968–973, 2008.

[113] J. Tighe and S. Lazebnik. Understanding scenes on many levels. In *Computer Vision. ICCV. IEEE International Conference on*, pages 335–342, nov. 2011.

[114] A. Torralba and A. Oliva. Statistics of natural image categories. *Network: Computation in Neural Systems*, 14(3):391–412, 2003.

[115] A. Torralba, K. Murphy, W. Freeman, and M. Rubin. Context-based vision system for place and object recognition. In *Computer Vision. ICCV. Ninth IEEE International Conference on*, volume 1, pages 273–280, Oct. 2003.

[116] 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. ISSN 0001-0782.

[117] 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.

[118] N. X. Vinh, J. Epps, and J. Bailey. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *The Journal of Machine Learning Research*, 9999:2837–2854, 2010.

[119] U. Von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007.

[120] H. Wallach, D. Mimno, and A. McCallum. Rethinking LDA: Why priors matter. *Advances in Neural Information Processing Systems*, 22:1973–1981, 2009.

[121] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. Locality-constrained linear coding for image classification. In *Computer Vision and Pattern Recognition. CVPR., IEEE Conference on*, pages 3360–3367. IEEE, 2010.

[122] X. Wang and E. Grimson. Spatial latent Dirichlet allocation. *Advances in Neural Information Processing Systems*, 20:1577–1584, 2007.

[123] M. Welling and K. Kurihara. Bayesian k-means as a maximization-expectation algorithm. In *Sixth SIAM International Conference on Data Mining*, volume 22, 2006.

[124] S. B. Williams, O. Pizarro, M. Jakuba, and N. Barrett. AUV benthic habitat mapping in South Eastern Tasmania. In A. Howard, K. Iagnemma, and A. Kelly, editors, *Field and Service Robotics*, volume 62 of *Springer Tracts in Advanced Robotics*, pages 275–284. Springer Berlin Heidelberg, 2010. ISBN 978-3-642-13407-4.

[125] 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, and A. Friedman. Monitoring of benthic reference sites: using an autonomous underwater vehicle. *Robotics Automation Magazine, IEEE*, 19(1):73–84, March 2012. ISSN 1070-9932.

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

[127] D. Wulsin, S. Jensen, and 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.

[128] 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. IEEE Conference on*, pages 1794–1801, 2009.

[129] K. Yu, T. Zhang, and Y. Gong. Nonlinear learning using local coordinate coding. In *Advances in Neural Information Processing Systems*, volume 22, pages 2223–2231, 2009.

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

[131] J. Zhang and C. Zhang. Multitask Bregman clustering. *Neurocomputing*, 74 (10):1720–1734, 2011.

[132] B. Zhao, L. Fei-Fei, and E. Xing. Image segmentation with topic random field. In K. Daniilidis, P. Maragos, and N. Paragios, editors, *Computer Vision – ECCV 2010*, volume 6315 of *Lecture Notes in Computer Science*, pages 785–798. Springer Berlin Heidelberg, 2010. ISBN 978-3-642-15554-3.

[133] M. Zhou, H. Chen, J. Paisley, L. Ren, G. Sapiro, and L. Carin.
Non-parametric Bayesian dictionary learning for sparse image representations.
Advances in Neural Information Processing, pages 1–9, 2009.

[134] O. Zobay. Mean field inference for the Dirichlet process mixture model.
Electronic Journal of Statistics, 3:507–545, 2009.

Appendix A

Some Useful Distributions and Expectations for Variational Bayes

In this appendix the expectations for the variational Bayes (VB) updates in the thesis are detailed. By no means is this an exhaustive list, only the distributions used in this thesis are presented. For more information on these distributions, and many more, good references are Bishop [12, ch. 2 & app. B] and Wikipedia.

A.1 Exponential Family

The exponential family likelihood has the form,

$$p(\mathbf{x}_n|\theta_k) = f(\mathbf{x}_n)g(\theta_k)\exp\{\boldsymbol{\phi}(\theta_k)^\top \mathbf{u}(\mathbf{x}_n)\}, \tag{A.1}$$

with the prior over the parameter,

$$p(\theta_k|\eta,\boldsymbol{\nu}) = h(\eta,\boldsymbol{\nu})g(\theta_k)^\eta \exp\{\boldsymbol{\phi}(\theta_k)^\top \boldsymbol{\nu}\}. \tag{A.2}$$

See `http://en.wikipedia.org/wiki/Exponential_family` for more information on this family of distributions.

A.1.1 Expectations over the likelihood

The expected log-likelihood is,

$$\mathbb{E}_{q_\theta}[\log p(\mathbf{x}_n|\theta_k)] = \log f(\mathbf{x}_n) + \mathbb{E}_{q_\theta}[\log g(\theta_k)] + \mathbb{E}_{q_\theta}[\boldsymbol{\phi}(\theta_k)]^\top \mathbf{u}(\mathbf{x}_n)\,, \tag{A.3}$$

A.1.2 Variational updates

The posterior variational hyper-parameters are,

$$\tilde{\eta}_k = \eta + \sum_{n=1}^{N} q(z_n = k)\,, \tag{A.4}$$

$$\tilde{\boldsymbol{\nu}}_k = \boldsymbol{\nu} + \sum_{n=1}^{N} q(z_n = k)\,\mathbf{u}(\mathbf{x}_n)\,. \tag{A.5}$$

A.1.3 Free energy expectations

The expectations of the model complexity penalty terms are,

$$\mathbb{E}_{q_\theta}\left[\log \frac{q(\theta_k)}{p(\theta_k|\eta,\boldsymbol{\nu})}\right] = (\tilde{\eta}_k - \eta)\,\mathbb{E}_{q_\theta}[\log g(\theta_k)] + \mathbb{E}_{q_\theta}[\boldsymbol{\phi}(\theta_k)]^\top(\tilde{\boldsymbol{\nu}}_k - \boldsymbol{\nu}) + \log h(\tilde{\eta}_k, \tilde{\boldsymbol{\nu}}_k) - \log h(\eta, \boldsymbol{\nu}), \tag{A.6}$$

A.2 Dirichlet Distribution

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

$$\mathrm{Cat}(z_n|\boldsymbol{\pi}) = \prod_{k=1}^{K} \pi_k^{\mathbf{1}[z_n=k]}. \tag{A.7}$$

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,

$$\mathrm{Dir}(\boldsymbol{\pi}|\alpha_1,\ldots,\alpha_K) = \frac{\Gamma(\sum_k \alpha_k)}{\prod_k \Gamma(\alpha_k)} \prod_{k=1}^{K} \pi_k^{\alpha_k - 1}, \tag{A.8}$$

here $\Gamma(\cdot)$ is a Gamma function. A symmetric Dirichlet prior, which is used often in the thesis, simply has $\alpha_k = \alpha \; \forall k$ or also commonly used is $\alpha_k = \alpha/K \; \forall k$.

A.2.1 Expectations over the likelihood

The log Categorical expectation under a generalised Dirichlet is,

$$\begin{aligned} \mathbb{E}_{q_{\boldsymbol{\pi}}}[\log p(z_n = k|\boldsymbol{\pi})] &= \mathbb{E}_{q_\pi}[\log \pi_k] \\ &= \Psi(\tilde{\alpha}_k) - \Psi\left(\sum_k \tilde{\alpha}_k\right), \end{aligned} \tag{A.9}$$

where $\Psi(\cdot)$ is a Digamma function.

A.2.2 Variational updates

The variational posterior hyper-parameter updates are,

$$\tilde{\alpha}_k = \alpha_k + \sum_{n=1}^{N} q(z_n = k)\,, \tag{A.10}$$

or for a symmetric Dirichlet as used in the thesis,

$$\tilde{\alpha}_k = \alpha + \sum_{n=1}^{N} q(z_n = k)\,. \tag{A.11}$$

A.2.3 Free energy expectations

The expectations of the model complexity penalty terms are,

$$\mathbb{E}_{q_{\boldsymbol{\pi}}}\left[\log \frac{q(\boldsymbol{\pi})}{\mathrm{Dir}(\boldsymbol{\pi}|\alpha_1,\ldots,\alpha_K)}\right] = \log\Gamma\left(\sum_{k=1}^{K}\tilde{\alpha}_k\right) - \log\Gamma\left(\sum_{k=1}^{K}\alpha_k\right) - \sum_{k=1}^{K}\log\Gamma(\tilde{\alpha}_k) + \sum_{k=1}^{K}\log\Gamma(\alpha_k) + \sum_{k=1}^{K}(\tilde{\alpha}_k - \alpha_k)\,\mathbb{E}_{q_\pi}[\log\pi_k]\,, \quad \text{(A.12)}$$

where $\mathbb{E}_{q_\pi}[\log\pi_k]$ is from Equation A.9.

A.3 Generalised Dirichlet Distribution

The Categorical distribution is most often used as the likelihood of the Generalised Dirichlet distribution in this thesis, see Equation A.7. The generalised Dirichlet prior on the mixture weights, $\mathrm{GDir}(\boldsymbol{\pi}_j|\mathbf{a},\mathbf{b})$, is similar to a truncated stick-breaking process [59],

$$\pi_k = v_k\prod_{l=1}^{k-1}(1-v_l), \qquad v_k \sim \begin{cases} \mathrm{Beta}(a_k,b_k) & \text{if } k<K \\ 1 & \text{if } k=K, \end{cases} \quad \text{(A.13)}$$

where $v_k \in [0,1]$ are "stick-lengths" for each group, and $\mathrm{Beta}(\cdot)$ is a Beta distribution,

$$\mathrm{Beta}(v_k|a_k,b_k) = \frac{\Gamma(a_k+b_k)}{\Gamma(a_k)\,\Gamma(b_k)} v_k^{a_k-1}\,(1-v_k)^{b_k-1}\,. \quad \text{(A.14)}$$

Here $\Gamma(\cdot)$ is a Gamma function.

A.3.1 Expectations over the likelihood

The log Categorical expectation under a generalised Dirichlet is,

$$\mathbb{E}_{q_{\boldsymbol{\pi}}}[\log p(z_n=k|\boldsymbol{\pi})] = \mathbb{E}_{q_\pi}[\log\pi_k]$$

$$= \mathbb{E}_{q_v}[\log v_k] + \sum_{l=1}^{k-1} \mathbb{E}_{q_v}[\log(1 - v_l)]\,, \tag{A.15}$$

where,

$$\mathbb{E}_{q_v}[\log v_k] = \begin{cases} \Psi(\tilde{a}_k) - \Psi\left(\tilde{a}_k + \tilde{b}_k\right) & \text{if } k < K \\ 0 & \text{if } k = K, \end{cases} \tag{A.16}$$

and,

$$\mathbb{E}_{q_v}[\log(1 - v_k)] = \Psi\left(\tilde{b}_k\right) - \Psi\left(\tilde{a}_k + \tilde{b}_k\right) \quad \text{if } k < K. \tag{A.17}$$

Here $\Psi(\cdot)$ is a Digamma function.

A.3.2 Variational updates

The variational posterior generalised Dirichlet hyper-parameters are,

$$\tilde{a}_k = a_k + \sum_{n=1}^{N} q(z_n = k)\,, \tag{A.18}$$

$$\tilde{b}_k = b_k + \sum_{n=1}^{N} \sum_{l=k+1}^{K} q(z_n = l)\,. \tag{A.19}$$

A.3.3 Free energy expectations

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

$$\mathbb{E}_{q_{\boldsymbol{\pi}}}\left[\log \frac{q(\boldsymbol{\pi})}{\mathrm{GDir}(\boldsymbol{\pi}|\mathbf{a}, \mathbf{b})}\right] = \sum_{k=1}^{K-1} \mathbb{E}_{q_\pi}\left[\log \frac{q(\pi_k)}{p(\pi_k|a_k, b_k)}\right], \tag{A.20}$$

where

$$\begin{aligned} \mathbb{E}_{q_\pi}\left[\log \frac{q(\pi_k)}{p(\pi_k|a_k, b_k)}\right] = (\tilde{a}_k - a_k)\,\mathbb{E}_{q_v}[\log v_k] + \left(\tilde{b}_k - b_k\right)\mathbb{E}_{q_v}[\log(1 - v_k)] \\ - \log\Gamma(\tilde{a}_k) + \log\Gamma(a_k) - \log\Gamma\left(\tilde{b}_k\right) + \log\Gamma(b_k) \\ + \log\Gamma\left(\tilde{a}_k + \tilde{b}_k\right) - \log\Gamma(a_k + b_k)\,. \end{aligned} \tag{A.21}$$

The free energy penalty term over the weights in Equation A.21 only sums to $K-1$ (degrees of freedom).

A.4 Gaussian-Wishart Distribution

Gaussian distributions are often used to describe clusters in this thesis, which take the form,

$$\mathcal{N}\left(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k^{-1}\right) = \frac{|\boldsymbol{\Lambda}_k|^{1/2}}{(2\pi)^{D/2}} \exp\left\{-\frac{1}{2}\left(\mathbf{x}_n - \boldsymbol{\mu}_k\right)^\top \boldsymbol{\Lambda}_k \left(\mathbf{x}_n - \boldsymbol{\mu}_k\right)\right\}. \tag{A.22}$$

A Gaussian-Wishart prior is placed over the parameters,

$$\mathcal{N}\left(\boldsymbol{\mu}_k|\mathbf{m}, (\gamma\boldsymbol{\Lambda}_k)^{-1}\right) = \frac{|\gamma\boldsymbol{\Lambda}_k|^{1/2}}{(2\pi)^{D/2}} \exp\left\{-\frac{\gamma}{2}\left(\boldsymbol{\mu}_k - \mathbf{m}\right)^\top \boldsymbol{\Lambda}_k \left(\boldsymbol{\mu}_k - \mathbf{m}\right)\right\}, \tag{A.23}$$

$$\mathcal{W}(\boldsymbol{\Lambda}_k|\boldsymbol{\Omega}, \rho) = \frac{|\boldsymbol{\Lambda}_k|^{(\rho-D-1)/2}}{2^{\rho D/2}\,|\boldsymbol{\Omega}|^{\rho/2}\,\Gamma_D\left(\frac{\rho}{2}\right)} \exp\left\{-\frac{1}{2}\mathrm{Tr}\left(\boldsymbol{\Omega}^{-1}\boldsymbol{\Lambda}_k\right)\right\}, \tag{A.24}$$

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

$$\Gamma_D\left(\frac{\rho}{2}\right) = \pi^{D(D-1)/4} \prod_{d=1}^{D} \Gamma\left(\frac{\rho + 1 - d}{2}\right), \tag{A.25}$$

and $\Gamma(\cdot)$ is a Gamma function.

A.4.1 Expectations over the likelihood

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

$$\mathbb{E}_{q_{\boldsymbol{\mu},\boldsymbol{\Lambda}}}\left[\log \mathcal{N}\left(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k^{-1}\right)\right] = \frac{1}{2}\mathbb{E}_{q_{\boldsymbol{\Lambda}}}[\log|\boldsymbol{\Lambda}_k|] - \frac{D}{2\tilde{\gamma}_k} - \frac{\tilde{\rho}_k}{2}\left(\mathbf{x}_n - \tilde{\mathbf{m}}_k\right)^\top \tilde{\boldsymbol{\Omega}}_k \left(\mathbf{x}_n - \tilde{\mathbf{m}}_k\right), \tag{A.26}$$

where

$$\mathbb{E}_{q_{\boldsymbol{\Lambda}}}[\log|\boldsymbol{\Lambda}_k|] = \sum_{d=1}^{D} \Psi\left(\frac{\tilde{\rho}_k + 1 - d}{2}\right) + D\log 2 + \log\left|\tilde{\boldsymbol{\Omega}}_k\right|, \tag{A.27}$$

and $\Psi(\cdot)$ is a Digamma function.

A.4.2 Variational updates

The variational posterior Gaussian-Wishart hyper-parameters are,

$$\tilde{\gamma}_k = \gamma + N_k, \tag{A.28}$$

$$\tilde{\mathbf{m}}_k = \frac{1}{\tilde{\gamma}_k} \left(\gamma \mathbf{m} + N_k \bar{\mathbf{x}}_k\right), \tag{A.29}$$

$$\tilde{\rho}_k = \rho + N_k, \tag{A.30}$$

$$\tilde{\boldsymbol{\Omega}}_k^{-1} = \boldsymbol{\Omega}^{-1} + N_k \mathbf{R}_k + \frac{\gamma N_k}{\tilde{\gamma}_k} (\bar{\mathbf{x}}_k - \mathbf{m})(\bar{\mathbf{x}}_k - \mathbf{m})^\top, \tag{A.31}$$

where

$$N_k = \sum_{n=1}^{N} q(z_n = k), \tag{A.32}$$

$$\bar{\mathbf{x}}_k = \frac{1}{N_k} \sum_{n=1}^{N} q(z_n = k)\, \mathbf{x}_n, \tag{A.33}$$

$$\mathbf{R}_k = \frac{1}{N_k} \sum_{n=1}^{N} q(z_n = k)\, (\mathbf{x}_n - \bar{\mathbf{x}}_k)(\mathbf{x}_n - \bar{\mathbf{x}}_k)^\top. \tag{A.34}$$

Note that $\rho \geq D - 1$.

A.4.3 Free energy expectations

The expectations of the model complexity penalty terms are,

$$\begin{aligned}
\mathbb{E}_{q_{\boldsymbol{\mu},\boldsymbol{\Lambda}}}\left[\log \frac{q(\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k)}{\mathcal{N}\left(\boldsymbol{\mu}_k | \mathbf{m}, (\gamma \boldsymbol{\Lambda}_k)^{-1}\right) \mathcal{W}(\boldsymbol{\Lambda}_k | \boldsymbol{\Omega}, \rho)}\right] &= \frac{D}{2}\left(\frac{\gamma}{\tilde{\gamma}_k} - \log \frac{\gamma}{\tilde{\gamma}_k} - \tilde{\rho}_k - 1\right) \\
&+ \frac{\rho}{2}\left(\log |\boldsymbol{\Omega}| - \log |\tilde{\boldsymbol{\Omega}}_k|\right) + \frac{\tilde{\rho}_k}{2} \mathrm{Tr}\left(\boldsymbol{\Omega}^{-1} \tilde{\boldsymbol{\Omega}}_k\right) + \frac{\tilde{\rho}_k \gamma}{2} (\tilde{\mathbf{m}}_k - \mathbf{m})^\top \tilde{\boldsymbol{\Omega}}_k (\tilde{\mathbf{m}}_k - \mathbf{m}) \\
&+ \sum_{d=1}^{D} \left(\frac{N_k}{2} \Psi\left(\frac{\tilde{\rho}_k + 1 - d}{2}\right) + \log \Gamma\left(\frac{\rho + 1 - d}{2}\right) - \log \Gamma\left(\frac{\tilde{\rho}_k + 1 - d}{2}\right)\right).
\end{aligned} \tag{A.35}$$

Appendix B

Functional Derivatives

Say we have a functional (function that takes function arguments),

$$I = \int F\left(x, y(x), y'(x)\right) dx$$

that takes arguments x, $y(x)$, and $y'(x)$ that we wish to optimise with respect to the function y, i.e. $\frac{\partial I}{\partial y} = 0$. We can do so using the *Euler-Lagrange* equation,

$$\frac{\partial I}{\partial y} = \frac{d}{dx} \cdot \frac{\partial F}{\partial y'} - \frac{\partial F}{\partial y} = 0. \tag{B.1}$$

For more information, see Collins [35, Ch. 11].