PARTIAL MEMBERSHIP LATENT DIRICHLET ALLOCATION

A Dissertation presented to

the Faculty of the Graduate School

at the University of Missouri

In Partial Fulfillment

of the Requirements for the Degree

Doctor of Philosophy

---

by

CHAO CHEN

Dr. Alina Zare, Dissertation Supervisor

MAY 2016

The undersigned, appointed by the Dean of the Graduate School, have examined the dissertation entitled:

PARTIAL MEMBERSHIP LATENT DIRICHLET ALLOCATION

Chao Chen,

a candidate for the degree of Doctor of Philosophy and hereby certify that, in their opinion, it is worthy of acceptance.

Dr. Alina Zare

Dr. James Keller

Dr. Marjorie Skubic

Dr. Dominic Ho

Dr. Mihail Popescu

ACKNOWLEDGMENTS

I would like to thank my advisor, Dr. Alina Zare, for all of her invaluable guidance, continued support and the numerous opportunities she provided me in my studies and research, without which this dissertation could not be realized. I would also like to thank my committee members, Dr. James Keller, Dr. Marjorie Skubic, Dr. Dominic Ho, and Dr. Mihail Popescu, for all of their help and valuable suggestions on improving this dissertation. Thank you to Dr. J. Tory Cobb of Naval Surface Warfare Center, for his support throughout this research. Thank you to my labmates and friends. I am particularly grateful to Changzhe Jiao, Matthew Cook, and Shanjie Chen for the thoughts, insights and discussion during my studies.

Thank you to my parents, Zhaofa Chen and Chuanlan He, my sister, Xue Chen, and all of my family for their continued love and support.

TABLE OF CONTENTS

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

ABSTRACT

For many years, topic models (e.g., pLSA, LDA, SLDA) have been widely used for segmenting and recognizing objects in imagery simultaneously. However, these models are confined to the analysis of categorical data, forcing a visual word to belong to one and only one topic. There are many images in which some regions cannot be assigned a crisp categorical label (e.g., transition regions between a foggy sky and the ground or between sand and water at a beach). In these cases, a visual word is best represented with partial memberships across multiple topics. To address this, a partial membership latent Dirichlet allocation (PM-LDA) model and associated parameter estimation algorithm are present.

PM-LDA defines a novel partial membership model for word and document generation. Different from the standard LDA model which assumes that each word belongs to one and only one topic, PM-LDA model allows words to have partial membership in multiple topics. This model can be useful for image/video documents where a visual word (an image patch) may be a mixture of multiple topics. For example, in a SONAR imagery where the gradually vanishing sand ripples blur the boundary between sand ripple region and flat sand region, it is impossible to tell where the sand ripple ends and the flat sand starts. In the proposed PM-LDA model, the visual words are represented with partial memberships in both "sand ripple" and "flat sand" topics, which is more reasonable than assigning them to one and only one topic as in the standard LDA model. A Gibbs sampling is employed for parameter estimation. Experimental results on simulated data, SONAR image dataset and natural image datasets show that PM-LDA can produce both crisp and soft semantic image segmentations; a capability existing methods do not have.

Chapter 1

Literature Review

1.1 Latent Dirichlet Allocation

Original topic models were used for discovering the topics hidden in document collections, thus facilitating the summary, organization, and search of large archives of texts [1]. Latent Dirichlet allocation (LDA) [2] is arguably the most popular and simplest topic model in application [3], which uncovers the latent topics and groups documents into latent topics based on a hierarchical Bayesian model.

LDA provides a generative probabilistic model for discrete datasets. In the context of text data, the dataset is a collection of D documents. Each document, a collection of words, is modeled as a finite mixture over a set of latent topics. For example, a bioinformatics article may have words drawn from the "genetics" topic and words drawn from the "data mining" topic, but no words from the "food" topic. Each topic is modeled as a multinomial distribution over words. For example, the "genetics" topic has words about genetics with

high probability and the "data mining" topic has words related to data mining with high probability [1]. Figure 1.1 shows three example topics derived from the TASA corpus [4]. In LDA, these hidden topics are inferred based on word co-occurrence; topics are formed by grouping co-occurring words into one topic.

word	prob.
DRUGS	.069
DRUG	.060
MEDICINE	.027
EFFECTS	.026
BODY	.023
MEDICINES	.019
PAIN	.016
PERSON	.016
MARIJUANA	.014
LABEL	.012
ALCOHOL	.012
DANGEROUS	.011
ABUSE	.009
EFFECT	.009
KNOWN	.008
PILLS	.008

(a) Topic 247

word	prob.
RED	.202
BLUE	.099
GREEN	.096
YELLOW	.073
WHITE	.048
COLOR	.048
BRIGHT	.030
COLORS	.029
ORANGE	.027
BROWN	.027
PINK	.017
LOOK	.017
BLACK	.016
PURPLE	.015
CROSS	.011
COLORED	.009

(b) Topic 5

word	prob.
MIND	.081
THOUGHT	.066
REMEMBER	.064
MEMORY	.037
THINKING	.030
PROFESSOR	.028
FELT	.025
REMEMBERED	.022
THOUGHTS	.020
FORGOTTEN	.020
MOMENT	.020
THNIK	.019
THING	.016
WONDER	.014
FORGET	.012
RECALL	.012

(c) Topic 43

Figure 1.1: An illustration of three topics extracted from the TASA corpus.

1.1.1 Generative Process

As a generative clustering method, LDA also defines a generative process of how the data is generated and structured. In LDA, the generative process for each document $\mathbf{w}_d$ in a corpus $\mathbf{D}$ is

1. For each document, $\mathbf{w}_d$, draw topic proportions $\pi_d \sim \text{Dir}(\alpha)$.

2. For each word

   (a) Draw topic assignment $z_{d,n} \sim \text{Mult}(\pi_d)$.

   (b) Draw word $w_{d,n} \sim \text{Mult}(\beta_{z_{d,n}})$.

Step 1 reflects the assumption that each document is consist of topics with different proportions. Step 2.(b) shows that each individual word is drawn from a topic determined in Step 2.(a), and that the topic is characterized as a multinomial distribution over words. A graphical model describing this generative process is shown in Figure 1.2. The shaded nodes, $w_{d,n}$, are the observable variables, and the transparent nodes, $\{\pi_d, z_{d,n}\}$, are latent variables.

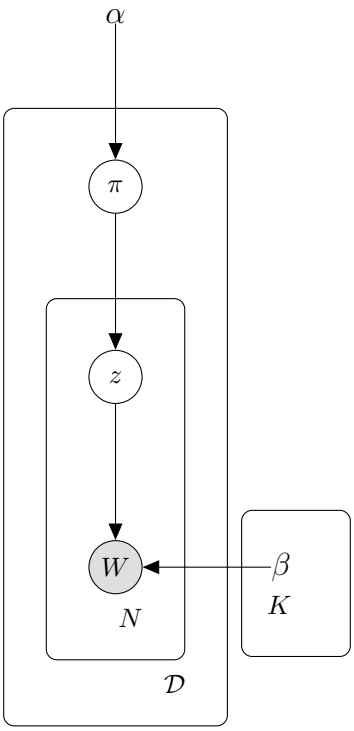


Figure 1.2: Graphical model of LDA.

1.1.2 Inference

For inference, the central problem for LDA is determining the posterior distribution of the hidden variables given the document,

$$p(\pi, \mathbf{z}|\mathbf{w}, \alpha, \beta) = \frac{p(\pi, \mathbf{z}, \mathbf{w}|\alpha, \beta)}{p(\mathbf{w}|\alpha, \beta)}. \tag{1.1}$$

The numerator can be decomposed into a hierarchy by examining the graphical model in Figure 1.2,

$$\begin{aligned} p(\pi, \mathbf{z}, \mathbf{w}|\alpha, \beta) &= p(\mathbf{w}|\mathbf{z}, \beta)p(\mathbf{z}|\pi)p(\pi|\alpha) \\ &= \prod_{n=1}^{N} \beta_{z_n, w_n} \prod_{n=1}^{N} \pi_{z_n} p(\pi|\alpha) \\ &= \left(\frac{\Gamma(\sum_{i=1}^{K} \alpha_i)}{\prod_{i=1}^{K} \Gamma(\alpha_i)} \prod_{i=1}^{K} {\pi_i}^{\alpha_i - 1} \right) \prod_{n=1}^{N} \prod_{i=1}^{K} \prod_{j=1}^{V} (\pi_i \beta_{i,j})^{{w_n}^j {z_n}^i}. \end{aligned} \tag{1.2}$$

By marginalizing over π and $\mathbf{z}$, the denominator can be expressed as

$$p(\mathbf{w}|\alpha, \beta) = \frac{\Gamma(\sum_{i=1}^{K} \alpha_i)}{\prod_{i=1}^{K} \Gamma(\alpha_i)} \int \left(\prod_{i=1}^{K} {\pi_i}^{\alpha_i - 1} \right) \left(\prod_{n=1}^{N} \sum_{i=1}^{K} \prod_{j=1}^{V} (\pi_i \beta_{i,j})^{{w_n}^j} \right) d\pi. \tag{1.3}$$

However, computing Equation (1.3) is intractable due to the coupling between π and β in the summation operation. In other words, the log of Equation (1.3) is unable to separate π and β. Since computing the exact posterior is intractable, several methods have been developed for performing approximate inference. Two commonly used approximate inference techniques used for LDA are collapsed Gibbs sampling [5] and variational inference (as used in the original LDA paper) [2].

Collapsed Gibbs Sampling

The LDA is interested in the topic assignment for each word $z_{d,n}$, the topic proportion π_d, and the topic-word distribution β. It's noted that both π_d and β can be calculated using $z_{d,n}$. Therefore, in the LDA model, the parameters of the multinomial distributions, π_d and β can be integrated out and the latent topic assignment $z_{d,n}$ will be simply sampled. This is called a collapsed Gibbs sampler. As shown in [5], to apply the collapsed Gibbs sampler, the probability of a topic k being assigned to a word $w_{d,n}$, given all other topic assignments to all other words, $\mathbf{z}_{-d,n}$, is needed and defined as

$$\begin{aligned} p(z_{d,n} = k|\mathbf{z}_{-d,n}, \mathbf{w}) &\propto p(z_{d,n} = k, \mathbf{z}_{-d,n}, \mathbf{w}) \\ &= p(w_{d.n}|z_{d,n} = k, \mathbf{z}_{-d,n}, \mathbf{w}_{-d,n})p(z_{d,n} = k|\mathbf{z}_{-d,n}, \mathbf{w}_{-d,n}) \\ &= p(w_{d.n}|z_{d,n} = k, \mathbf{z}_{-d,n}, \mathbf{w}_{-d,n})p(z_{d,n} = k|\mathbf{z}_{-d,n}), \end{aligned} \tag{1.4}$$

where the first term $p(w_{d.n}|z_{d,n} = k, \mathbf{z}_{-d,n}, \mathbf{w}_{-d,n})$ is like the likelihood and the second term $p(z_{d,n} = k|\mathbf{z}_{-d,n})$ is like a prior.

The first term,

$$\begin{aligned} & p(w_{d.n}|z_{d,n} = k, \mathbf{z}_{-d,n}, \mathbf{w}_{-d,n}) \\ = & \int p(w_{d,n}|z_{d,n} = k, \beta^{(k)})p(\beta^{(k)}|\mathbf{z}_{-d,n}, \mathbf{w}_{-d,n})d\beta^{(k)} \\ = & \int p(w_{d,n}|\beta^{(k)})p(\beta^{(k)}|\mathbf{z}_{-d,n}, \mathbf{w}_{-d,n})d\beta^{(k)}, \end{aligned} \tag{1.5}$$

where

$$\begin{aligned} p(\beta^{(k)}|\mathbf{z}_{-d,n}, \mathbf{w}_{-d,n}) &\propto p(\mathbf{w}_{-d,n}|\beta^{(k)}, \mathbf{z}_{-d,n})p(\beta^{(k)}) \\ &\sim \text{Dir}(C_{-n,k}^{(w_{d,n})} + \beta). \end{aligned} \tag{1.6}$$

Then, Equation (1.5) becomes the expectation of the Dirichlet distribution in Equation (1.6),

$$p(w_{d.n}|z_{d,n} = k, \mathbf{z}_{-d,n}, \mathbf{w}_{-d,n}) = \frac{C_{-n,k}^{(w_{d,n})} + \beta}{C_{-n,k}^{(.)} + V\beta}. \tag{1.7}$$

For the second term,

$$p(z_{d,n} = k|\mathbf{z}_{-d,n}) = \int p(z_{d,n} = k|\pi_d)p(\pi_d|\mathbf{z}_{-d,n})d\pi_d, \tag{1.8}$$

where

$$\begin{aligned} p(\pi_d|\mathbf{z}_{-d,n}) &\propto p(\mathbf{z}_{-d,n}|\pi_d)p(\pi_d) \\ &\sim \text{Dir}(C_{-n,k}^{(d)} + \alpha). \end{aligned} \tag{1.9}$$

Then, Equation (1.8) becomes the expectation of the Dirichlet distribution in Equation (1.9),

$$p(z_{d,n} = k|\mathbf{z}_{-d,n}) = \frac{C_{-n,k}^{(d)} + \alpha}{C_{-n}^{(d)} + K\alpha}. \tag{1.10}$$

Thus Equation (1.4) can be expressed as,

$$p(z_{d,n} = k|\mathbf{z}_{-d,n}, \mathbf{w}) \propto \frac{C_{-n,k}^{(w_{d,n})} + \beta}{C_{-n,k}^{(.)} + V\beta} \frac{C_{-n,k}^{(d)} + \alpha}{C_{-n}^{(d)} + K\alpha}, \tag{1.11}$$

which involves four count variables,

- the topic-term count, $C_{-n,k}^{(w_{d,n})}$, the number of word $w_{d,n}$ assigned to topic k in a document,
- the topic-term sum, $C_{-n,k}^{(.)}$, the total number of words assigned to topic k,
- the document-topic count, $C_{-n,k}^{(d)}$, the number of words assigned to topic k excluding the current one, and
- the document-topic sum, $C_{-n}^{(d)}$, the total number of topics assigned to document d excluding the current one.

The first ratio expresses the probability of word w under topic k, and the second ratio expresses the probability of topic k in document d [5].

Based on Equation (1.6) and (1.9), the topic-word distribution β and the topic proportion π are computed as following,

$$\beta_{k,w} = \frac{C_w^{(k)} + \beta}{\sum_{w=1}^{V} C_w^{(k)} + V\beta}, \tag{1.12}$$

$$\pi_d(k) = \frac{C_k^{(d)} + \alpha}{\sum_{k=1}^{K} C_k^{(d)} + K\alpha}, \tag{1.13}$$

where $C_w^{(k)}$ is the frequency of word w assigned to topic k, and $C_k^{(d)}$ is the number of words assigned to topic k.

Variational Inference

Another major approximate inference method used for LDA model is mean-field variational inference. The basic idea of variational inference is to first pick a family of distributions, $q(\pi, \mathbf{z}|\gamma, \phi)$, over the latent variables with its own variational parameters (γ, ϕ), and then

find the optimal variational parameters (γ^*, ϕ^*) that minimizes the difference between the variational distribution $q(\pi, \mathbf{z}|\gamma, \phi)$ and the true posterior $p(\pi, \mathbf{z}|\mathbf{w}, \alpha, \beta)$. Thus the inference problem is transformed to an optimization problem.

The mean-field variational method used for LDA inference approximates the true posterior $p(\pi, \mathbf{z}|\alpha, \beta)$ using a simpler and factored variational distribution, defined as,

$$q(\pi, \mathbf{z}|\gamma, \phi) = q(\pi|\gamma) \prod_{n=1}^{N} q(z_n|\phi_n), \tag{1.14}$$

where γ is the Dirichlet parameter and $(\phi_1, ..., \phi_N)$ are the multinomial parameters. This variational distribution is proposed by dropping the edges and nodes in the graphical model (Figure 1.2) that result in the problematic coupling between π and β, namely, the edges between π, $\mathbf{z}$, and $\mathbf{w}$, and the $\mathbf{w}$ nodes. The simplified graphical model for the variational distribution is shown in Figure 1.3.

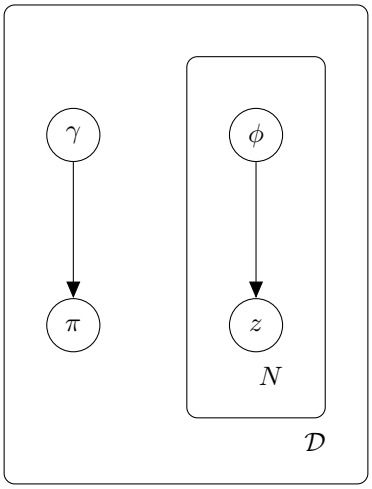


Figure 1.3: Graphical model of the variational distribution.

The optimization procedure to determine (γ^*, ϕ^*) is formalized as

$$(\gamma^*, \phi^*) = \underset{(\gamma,\phi)}{\arg\min}\, D_{KL}\left(q(\theta, \mathbf{z}|\gamma, \phi)||p(\theta, \mathbf{z}|\mathbf{w}, \alpha, \beta)\right), \tag{1.15}$$

with

$$\begin{aligned} & D_{KL}\left(q(\theta, \mathbf{z}|\gamma, \phi)||p(\theta, \mathbf{z}|\mathbf{w}, \alpha, \beta)\right) \\ = \;& E_q\left[\log q(\theta, \mathbf{z}|\gamma, \phi)\right] - E_q\left[\log p(\theta, \mathbf{z}|\mathbf{w}, \alpha, \beta)\right] \\ = \;& E_q\left[\log q(\theta, \mathbf{z}|\gamma, \phi)\right] - E_q\left[\log p(\theta, \mathbf{z}, \mathbf{w}|\alpha, \beta)\right] + \log p(\mathbf{w}|\alpha, \beta), \end{aligned} \tag{1.16}$$

where $D_{KL}\left(q||p\right)$ is the Kullback-Leibler (KL) divergence between the variational distribution q and the true posterior p.

The optimal values of (γ, ϕ) are found by minimizing $D_{KL}\left(q||p\right)$. Since the term $\log p(\mathbf{w}|\alpha, \beta)$ in D_{KL} does not depend on the variational distribution q, as a function of q, minimizing the KL divergence is equivalent to maximizing a lower bound L defined as

$$\begin{aligned} L(\gamma, \phi; \alpha, \beta) &= \log p(\mathbf{w}|\alpha, \beta) - D_{KL} \\ &= E_q\left[\log p(\theta, \mathbf{z}, \mathbf{w}|\alpha, \beta)\right] - E_q\left[\log q(\theta, \mathbf{z}|\gamma, \phi)\right]. \end{aligned} \tag{1.17}$$

This maximization is achieved using an iterative fixed-point method, yielding the following update equations:

$$\begin{aligned} \phi_{ni} &\propto \beta_{iw_n} \cdot e^{\Psi(\gamma_i) - \Psi\left(\sum_{j=1}^{K} \gamma_j\right)} \\ \gamma_i &= \alpha_i + \sum_{n=1}^{N} \phi_{ni}, \end{aligned}$$

where $\Psi(\cdot)$ is the digamma function, the first derivative of the log Gamma function. The variational inference procedure is summarized in Algorithm 1.

Algorithm 1 A variational inference algorithm for LDA

Initialize $\phi_{ni}^0 := 1/K$ for all i and n
(The probabilities of a word generated by different topics are equal.)
Initialize $\gamma_i := \alpha_i + N/K$ for all i
($\gamma_i = \alpha_i + \sum_{n=1}^{N} \phi_{ni} = \alpha + N/K$)
for $n \leftarrow 1$ to N **do**
  for $i \leftarrow 1$ to K **do**
    $\phi_{ni}^{t+1} := \beta_{iw_n} e^{(\Psi(\gamma_i{}^t))}$
    normalize ${\phi_n}^{t+1}$ to sum to 1.
    ${\gamma}^{t+1} := \alpha + \sum_{n=1}^{N} {\phi_n}^{t+1}$
  end for
end for

Although the observed document/words $\mathbf{w}$ is not explicitly expressed in $q(\pi, \mathbf{z}|\gamma, \phi)$, the variational distribution actually depends on the observed $\mathbf{w}$. Since the true posterior distribution $p(\pi, \mathbf{z}|\mathbf{w}, \alpha, \beta)$ varies with $\mathbf{w}$, the optimal (γ^*, ϕ^*) obtained from the optimization procedure is a function of $\mathbf{w}$. In this sense, the resulting variational distribution $q(\pi, \mathbf{z}|\gamma^*, \phi^*)$ can be considered as an approximation to the posterior distribution.

In the inference procedure, it's assumed that the model hyperparameters including the Dirichlet parameter α and the topics β are known. Thus certain tasks can be completed, such as analyzing the topic proportions in a given document and tracing each word back to the topic. However, in practice, these topics are not known in advance. They are believed to be hiding in the collected documents and can be uncovered through the parameter estimation.

The parameter estimation procedure is to find the hyperparameters $\{\alpha, \beta\}$ that maxi-

mize the log likelihood defined as,

$$\ell(\alpha, \beta) = \sum_{d=1}^{D} \log p(\mathbf{w}_d|\alpha, \beta). \tag{1.18}$$

This problem is often solved using the Expectation-Maximization (EM) algorithm where the variational parameters (γ_d^*, ϕ_d^*) and the hyperparameters $\{\alpha, \beta\}$ are updated iteratively. More formally, the proposed EM algorithm cycles iteratively between the following two steps:

- E-step: For each document $\mathbf{w}_d$, find the optimal variational parameters (γ_d^*, ϕ_d^*) by maximizing the lower bound L in Equation (1.17), assuming that the hyperparameters $\{\alpha, \beta\}$ are known. This step is the same as the variational inference. With these parameters, the expectation of the log likelihood of the complete data (Equation (1.18)) can be computed,

- M-step: Maximize the lower bound on the log likelihood with respect to $\{\alpha, \beta\}$. The updates for $\{\alpha, \beta\}$ are

  1. M-step update for β:

  $$\beta_{ij} \propto \sum_{d=1}^{D} \sum_{n=1}^{N_d} \phi_{dni} w_{dn}{}^{j}$$

  2. M-step update for α:

     The update for α is implemented using a linear-scaling Newton-Raphson method, with updates carried out in log-space (assuming a uniform α)

  $$\log(\alpha^{t+1}) = \log(\alpha^t) - \frac{\frac{dL}{d\alpha}}{\frac{d^2L}{d\alpha^2}\alpha + \frac{dL}{d\alpha}}, \tag{1.19}$$

with

$$\begin{aligned}\frac{dL}{d\alpha} &= D(K\Psi'(K\alpha) - K\Psi'(\alpha)) + \sum_{d=1}^{D}\left(\Psi(\gamma_{d,i}) - \Psi\left(\sum_{j=1}^{K}\gamma_{d,j}\right)\right), \\ \frac{d^2L}{d\alpha^2} &= D(K^2\Psi''(K\alpha) - K\Psi''(\alpha)).\end{aligned}$$

1.1.3 LDA Variations

The LDA model is a generative model for a collection of exchangeable discrete data [6]. With high modularity, it has been extended in many different ways.

Relation-based LDA Variants

One type of extension focuses on modeling relations between topics, documents, or even corpus. Considering that the presence of one latent topic may be correlated with the presence of another, the correlated topic model (CTM) replaces the Dirichlet distribution with the logistic normal distribution for the topic proportions, allowing for covariance structure among topics [7]. Relational topic model (RTM) assumes that each pair of documents is connected by a link, a binary random variable conditioned on the contents in the two documents. This model can be used to predict links and words for network data, such as social networks of friends and citation networks [8]. Markov Topic Models (MTMs) models the correlations of different corpora as Gaussian Markov random fields, which capture both the internal topic structure within each corpus and the relationships between topics across the corpora [9].

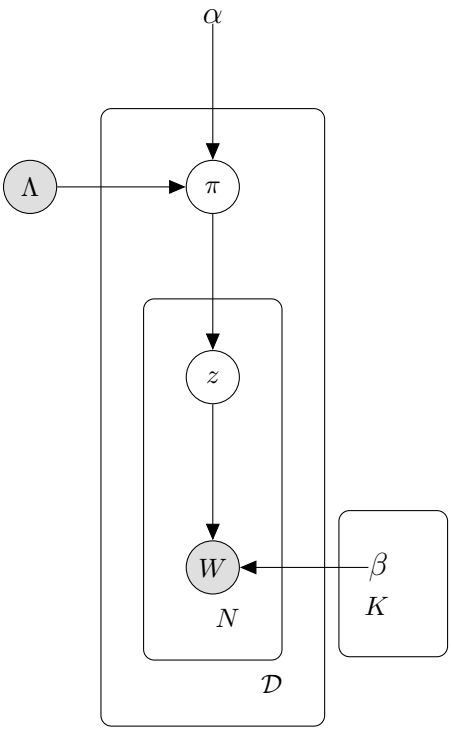
Dynamic LDA

The LDA model implicitly assumes that the documents are fully exchangeable. However, data are naturally collected along time [10]. This assumption is inappropriate for certain document collections, such as scholarly journals, news articles, that reflect topic evolution over time [7]. Dynamic mixture models (DMM) [10] assumes that the topic mixture proportion of each document is dependent on previous topic mixture proportions. A discrete-time dynamic topic model (dDTM) [7] defines a state space model for the dynamics of underlying topics and incorporates the state space model to LDA model. In dDTM, however, the time discretization ignores the time dependency of documents inside a period, which largely affect the resolution at which to fit the model [11]. Topic models with continuous time stamps have been proposed. Topics over times (TOT) [12] parameterizes a continuous distribution over time associated with each topic, and topics are responsible for generating both observed timestamps and words [12]. The dDTM model is further extended to a continuous time dynamic topic model (cDTM) [11] by replacing the discrete state space model with continuous Brownian motion.

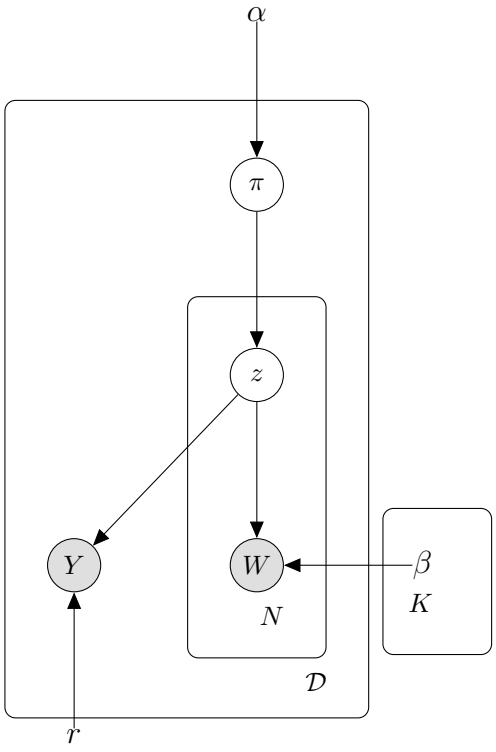
Supervision-based LDA Variants

For the original LDA model, the label assignment is at the word level where only the words in the documents will be assigned a label (topic) after the inference procedure [13]. Another important variation on LDA model is to label documents, pairing the document with a response (label). In supervised Latent Dirichlet Allocation (sLDA), the supervision is incorporated by attaching to each document an observed response variable y which is assumed to be a continuous random variable with a distribution as a generalized linear model parameterized by r [13]. In the generative process of sLDA, after generating a

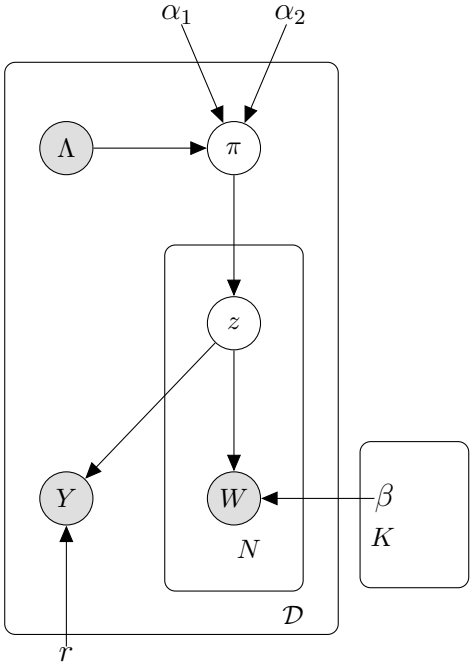
document as in LDA, a response variable y is drawn from a generalized linear model as the document label. MedLDA shares a very similar generative process with sLDA. It differs from sLDA in that, instead of learning a point estimate for r as in sLDA, MedLDA learns a distribution $q(r)$ in a max-margin manner. In MedLDA model, the response variable can be continuous or discrete, allowing for applicability of MedLDA in both regression and classification problem [14]. The graphical model of MedLDA is shown in Figure 1.4b. Labeled LDA (L-LDA) incorporates supervision by simply constraining the topic model to use only those topics that correspond to a document's (observed) label set. In the generative process, the topic proportions are generated only for observed topics, not for all the topics as in sLDA [15]. The graphical model of L-LDA is shown in Figure 1.4a. Doubly supervised LDA (DSLDA) integrates two types of supervision, document-level labels for topics (in L-LDA), and an overall category inferred form topics (in MedLDA). In DSLDA, the K topics are categorized into two types: K_1 latent topics that are never observed, and K_2 supervised topics that are observed in training but not in test data. Correspondingly $\alpha = (\alpha_1, \alpha_2)$ where α_1 is a parameter of a Dirichlet distribution of dimension K_1 and α_2 is a parameter of a Dirichlet distribution of dimension K_2. A document is assumed to contain certain latent topics and certain supervised topics. So the topic proportions for a document are generated only for those latent topics and those supervised topics present in the document [16]. The graphical model of DSLDA is shown in Figure 1.4c.



(a) Labeled LDA



(b) MedLDA



(c) Doubly Supervised LDA

Figure 1.4: Graphical models of L-LDA, MedLDA and DSLDA.

1.1.4 LDA-based Topic Models for Imagery

Topic models can be adapted to many kinds of data, such as genetic data, images, and social networks [1]. Some variations on topic models have been used for image/video data. For example, probabilistic latent semantic analysis (PLSA) [17] and LDA have been used to discover object categories from a collection of images [18, 19] and scene categories [20, 21]. In these works, images are treated as documents, with each image being represented as a collection of local patches. Each patch is assigned to the most similar visual word from a large vocabulary. The *visual words* are equivalent to words in text document, which are usually quantized image descriptors, e.g., SIFT descriptor. The vocabulary is established by densely extracting descriptors from the given images and then quantizing them into the *visual words* to form a vocabulary. The quantization is typically performed by k-means clustering with k as the predefined number of visual words in the vocabulary and the cluster centers as the visual words.

Visual Word Ambiguity

In the LDA model, an image document is modeled as a histogram of discrete visual words, also called "bag of visual words". Impressive results for image analysis and classification using the bag of words model have been demonstrated in [18–20, 22]. One inherent step of "bag of visual words" model is to build a dictionary of visual words and quantize a continuous feature descriptor to the nearest visual word in the visual dictionary. The underlying assumption of the quantization is that an image descriptor will be mapped to one and only one representative visual word (the nearest one). However, in practice, an image descriptor may not be close enough to any visual word in the dictionary, or may be close to multiple visual words. Thus an image descriptor may match to zero, or multiple visual word

candidates, resulting in the visual word ambiguity [23]. To accommodate the visual word ambiguity, Gemert, et al. [23] investigated four types of soft assignment of visual words to image descriptors and demonstrated the effectiveness of soft assignment in improving classification performance. Weinshall, et al. [24] incorporated the soft word assignment into the LDA model by adding into the generative process one additional step where the descriptor is sampled from the generative word model. The histogram of word counts was replaced by a histogram of pseudo word counts, and the parameter estimation depended on the average covariance matrix between pseudo counts.

Spatial Information

Another noticeable problem with the bag of visual words model is that each visual word is assumed to be drawn independently from its corresponding topic, thus the spatial structure/relationships among the visual words (e.g., image patches) are ignored. While the ignorance of spatial structure can greatly reduce the computational complexity, the spatial relationships among the image patches are often critical for solving many vision problems [20]. Li [20], Sudderth [25], and Wang et al. [26] extended the LDA model by considering the spatial locations of image patches to help the computer vision problems, such as the detection, recognition and classification of objects. The spatially coherent latent topic model (Spatial-LTM) [27] assumes that pixels should share the same latent topic assignment if they are in a neighboring region with similar appearance. To enforce the spatial coherence, in Spatial-LTM, an image is over segmented into regions of homogeneous appearances and only one topic label will be assigned to all local patches within the same region [27, 28]. The spatial Latent Dirichlet Allocation (SLDA) [26] encodes the spatial information by designing the document. It defines a generative procedure to assign words to documents

that words close in space have a high probability to be grouped into the same document. Different from Spatial-LTM that considers the spatial consistency between adjacent local patches, topic random field (TRF) [28] enforces the spatial coherence between adjacent over-segmented regions and defines a Markov Random Field over hidden topic assignment of super-pixels (regions) in an image.

1.2 Fuzzy Partitioning

Hard clustering methods partition a dataset by assigning each data point to exactly one cluster. No data point belongs to multiple clusters simultaneously; Clusters are mutually exclusive. Fuzzy methods, in contrast, allow a data point to belong to several clusters with different degrees of membership, and allow fuzzy clusters to be disjoint, or overlapping. Partial membership is a quite intuitive and practically useful idea [29].

1.2.1 Fuzzy C-Means

Fuzzy C-Means (FCM) [30] is a classical algorithm tackling the fuzzy clustering problem, which iteratively minimizes the objective function

$$J = \sum_{n=1}^{N}\sum_{k=1}^{K} z_{nk}^{m} d^2(\mathbf{x}_n, \boldsymbol{\mu}_k), \quad \text{s.t.} \quad \sum_{k=1}^{K} z_{nk} = 1, z_{nk} > 0 \tag{1.20}$$

with update equations defined as

$$z_{nk} = \frac{\left(\frac{1}{d^2(\mathbf{x}_n,\boldsymbol{\mu}_k)}\right)^{\frac{1}{m-1}}}{\sum_{k=1}^{K}\left(\frac{1}{d^2(\mathbf{x}_n,\boldsymbol{\mu}_k)}\right)^{\frac{1}{m-1}}}, \tag{1.21}$$

$$\boldsymbol{\mu}_k = \frac{\sum_{n=1}^{N} z_{nk}^m \mathbf{x}_n}{\sum_{n=1}^{N} z_{nk}^m}, \tag{1.22}$$

where $m > 1$ is the fuzzifier parameter, z_{nk} represents the degree of membership of data $\mathbf{x}_n$ to cluster k, $\boldsymbol{\mu}_k$ is the prototype of cluster k, and $d^2(\mathbf{x}_n, \boldsymbol{\mu}_k)$ is the squared distance between $\mathbf{x}_n$ and $\boldsymbol{\mu}_k$. The fuzzifier m controls the degrees of partial membership, and $d^2(\mathbf{x}_n, \boldsymbol{\mu}_k)$ control the type of clusters.

The choice of distance measure $d^2(\mathbf{x}_n, \boldsymbol{\mu}_k)$ determines the retrieved geometry of feature space. The original FCM used Euclidean distance

$$d^2(\mathbf{x}_n, \boldsymbol{\mu}_k) = (\mathbf{x}_n - \boldsymbol{\mu})^T(\mathbf{x}_n - \boldsymbol{\mu})$$

to produce hyper-spherical structural clusters. Extensions [31, 32] considered the Mahalanobis distance

$$d^2(\mathbf{x}_n, \boldsymbol{\mu}_k) = (\mathbf{x}_n - \boldsymbol{\mu}_k)^T\Sigma_k^{-1}(\mathbf{x}_n - \boldsymbol{\mu}_k),$$

assuming the clusters to be hyper-ellipsoidal. The Gustafson-Kessel algorithm (GK) [31] searches for ellipsoidal clusters with approximately the same size and the Gath and Geva algorithm(GG) [32] searches for ellipsoidal clusters of varying size [33]. These methods have been traditionally used to find "compact" or "filled" clusters [34]. To detect hollow or shell-like clusters, a large number of fuzzy shell clustering algorithms have been proposed, which used different kinds of cluster prototypes and different distance measures

[33]. The fuzzy C shells (FCS) algorithm [35, 36] utilizes hyper-spherical-shells as cluster prototypes. The fuzzy C ellipsoidal shells algorithm (FCES) [37] utilizes hyper-ellipsoidal-shells as cluster prototypes. The fuzzy C quadric shells algorithm (FCQS) [38] utilizes hyper-quadric-shells as cluster prototypes [33].

Fuzzy clustering techniques have been widely used in image processing and computer vision. Pham et al. [39] proposed an Adaptive FCM (AFCM) algorithm for segmentation of Magnetic resonance images that have been corrupted by shading effects. Ahmed et al. [40] developed a FCM_S algorithm for MRI data segmentation, which incorporated spatial constraints to increase the robustness of FCM to noise. Chen and Zhang [41] proposed several variants of FCM_S for robust image segmentation, which simplified the computation of FCM_S and enhanced its robustness to noise and outliers. Delbo et al. [42] applied FCQS to hyperbolic signatures recognition in subsurface radar images.

1.2.2 Bayesian Fuzzy Clustering

Glenn et al. [43] proposed a Bayesian Fuzzy Clustering (BFC) model and associated algorithms to bridge and extend probabilistic clustering and fuzzy clustering methods. In the BFC model, membership vectors are modeled as Dirichlet distributed random variables that indicate the degree of membership of each input data point within each cluster. Input data points are then modeled as Gaussian random variables whose natural parameters are convex combinations of the cluster natural parameters. The BFC model defines a data likelihood distribution (called the Fuzzy Data Likelihood (FDL)),

$$p(\mathbf{X}|\mathbf{Z},\mathbf{Y}) = \prod_{n=1}^{N} \frac{1}{Z(\mathbf{u}_n, m, \mathbf{Y})} \prod_{k=1}^{K} \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \mathbf{Q} = z_{nk}^m \mathbf{I})$$

with

$$\mathbf{Y} = \{\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_K\}, \tag{1.23}$$

and a prior distribution for the cluster membership (called Fuzzy Cluster Prior (FCP)),

$$\tilde{p}(\mathbf{Z}|Y) = \prod_{n=1}^{N} Z(\mathbf{u}_n, m, \mathbf{Y}) \left(\prod_{k=1}^{K} z_{nk}^{-mp/2} \right) \text{Dir}(\mathbf{z}_n|\boldsymbol{\alpha}) \tag{1.24}$$

and a Gaussian prior distribution on cluster prototypes

$$p(\mathbf{Y}) = \prod_{k=1}^{K} \mathcal{N}(\boldsymbol{\mu}_k|\boldsymbol{\mu}_y, \boldsymbol{\Sigma}_y), \tag{1.25}$$

where $Z(\mathbf{z}_n, m, \mathbf{Y})$ is a normalization constant, p is the dimensionality of the data, $\boldsymbol{\mu}_y$ and $\boldsymbol{\Sigma}_y$ are the sample mean and sample covariance, respectively. The joint likelihood of data and parameters

$$\begin{aligned} p(\mathbf{X}, \mathbf{Z}, \mathbf{Y}) &= p(\mathbf{X}|\mathbf{Z}, \mathbf{Y})\tilde{p}(\mathbf{Z}|\mathbf{Y})p(\mathbf{Y}) \\ &\propto \exp\left\{ -\frac{1}{2} \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk}^{m} \left(\mathbf{x}_n - \boldsymbol{\mu}_k\right)^T \left(\mathbf{x}_n - \boldsymbol{\mu}_k\right) \right\} \\ &\quad \prod_{n=1}^{N} \prod_{k=1}^{K} z_{nk}^{\alpha_k - 1} \times \exp\left\{ -\frac{1}{2} \sum_{k=1}^{K} \left(\boldsymbol{\mu}_k - \boldsymbol{\mu}_y\right)^T \boldsymbol{\Sigma}_y^{-1} \left(\boldsymbol{\mu}_k - \boldsymbol{\mu}_y\right) \right\}. \end{aligned} \tag{1.26}$$

Given the cluster prototype $\boldsymbol{\mu}_k$ and membership z_{nk}, the distribution of data point $\mathbf{x}_n$ is described as

$$p(\mathbf{x}_n|\mathbf{z}_n, \boldsymbol{\mu}_k) \propto \exp\left\{ -\frac{1}{2} \left[\left(\sum_{k=1}^{K} z_{nk}^{m} \right) \mathbf{x}_n^T \mathbf{x}_n - 2\mathbf{x}_n^T \left(\sum_{k=1}^{K} z_{nk}^{m} \boldsymbol{\mu}_k \right) + \left(\sum_{k=1}^{K} z_{nk}^{m} \boldsymbol{\mu}_k^T \boldsymbol{\mu}_k \right) \right] \right\}. \tag{1.27}$$

Let $\mathbf{Q} = \sum_{k=1}^{K} z_{nk}^{m} \mathbf{I}$, and $\boldsymbol{\mu} = \frac{1}{\sum_{k=1}^{K} z_{nk}^{m}} \left(\sum_{k=1}^{K} z_{nk}^{m} \boldsymbol{\mu}_k \right)$, Equation (1.27) can be rewrit-

ten as

$$
\begin{aligned}
p(\mathbf{x}_n|\mathbf{z}_n, \boldsymbol{\mu}_k) \quad &\propto \quad \exp\left\{-\frac{1}{2}\left[\mathbf{x}_n^T\mathbf{Q}\mathbf{x}_n - 2\mathbf{x}_n^T\mathbf{Q}\boldsymbol{\mu} + \boldsymbol{\mu}^T\mathbf{Q}\boldsymbol{\mu}\right]\right\} \\
&\times \quad \exp\left\{-\frac{1}{2}\left[\left(\sum_{k=1}^{K} z_{nk}^m \boldsymbol{\mu}_k^T\boldsymbol{\mu}_k\right) - \boldsymbol{\mu}^T\mathbf{Q}\boldsymbol{\mu}\right]\right\}, \qquad (1.28)
\end{aligned}
$$

then

$$
p(\mathbf{x}_n|\mathbf{z}_n, \boldsymbol{\mu}_k) = \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}, \mathbf{Q}). \qquad (1.29)
$$

For data point $\mathbf{x}_n$, the generative process implicitly defined in the BFC is as follows:

(a) Draw the membership vector $\mathbf{z}_n$ from the FCP.

(b) Draw a data point $\mathbf{x}_n$ from a Gaussian distribution, $\mathbf{x}_n \sim \mathcal{N}(|\boldsymbol{\mu}, \mathbf{Q})$, with

$$
\mathbf{Q} = \sum_{k=1}^{K} z_{nk}^m \mathbf{I} \qquad \text{and} \qquad \boldsymbol{\mu} = \frac{1}{\sum_{k=1}^{K} z_{nk}^m}\left(\sum_{k=1}^{K} z_{nk}^m \boldsymbol{\mu}_k\right).
$$

The FCP is an improper prior which is uninformative about the membership prior belief. It cannot be normalized over the interval $[0, 1]$. Thus, how to sampling memberships from FCP is unclear in [43]. It is also noted in [43] that the FCP can be converted to a proper prior by replacing the second term $\left(\prod_{k=1}^{K} z_{nk}^{-mp/2}\right)$ with a product of Inverse-Gamma distribution with shape parameter as $mp/2 - 1$ and scale parameter to be small.

1.2.3 Bayesian Partial Membership Model

Heller et al. [29] derived a Bayesian partial membership model (BPM) by extending the standard mixture model. The generative processes for both the BFC and BPM models are similar. The main difference between the BPM and BFC models is that the BFC uses both

a fixed *fuzzifier* parameter and a scaling parameter to control the degree of mixing between topics. In contrast, in the BPM, the degree of mixing between topics is controlled only through a scaling hyper-parameter, s, found in the prior distribution on partial membership values. This hyper-parameter is modeled as an exponential random variable.

In standard finite mixture model, the probability of a data point, $\mathbf{x}_n$ given its assignment indicator $\mathbf{z}_n$ and the cluster parameters $\mathbf{\Theta} = \{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, ..., \boldsymbol{\theta}_K\}$, is defined as

$$\begin{aligned} & p(\mathbf{x}_n|\mathbf{z}_n, \mathbf{\Theta}) = \prod_{k=1}^{K} p_k(\mathbf{x}_n|\boldsymbol{\theta}_k)^{z_{nk}}, \\ & z_{nk} \in \{0, 1\}, \\ & \sum_{k=1}^{K} z_{nk} = 1, \end{aligned} \tag{1.30}$$

where $\mathbf{z}_n = [z_{n1}, z_{n2}, ..., z_{nK}]$. If $z_{nk} = 1$, the data point $\mathbf{x}_n$ belongs to cluster k. For example, with $\mathbf{z}_n = [0, 0, 1, 0]$,

$$p(\mathbf{x}_n|\mathbf{z}_n, \mathbf{\Theta}) = p_1(\mathbf{x}_n|\boldsymbol{\theta}_1)^0 \cdot p_2(\mathbf{x}_n|\boldsymbol{\theta}_2)^0 \cdot p_3(\mathbf{x}_n|\boldsymbol{\theta}_3)^1 \cdot p_4(\mathbf{x}_n|\boldsymbol{\theta}_4)^0 \tag{1.31}$$

Standard finite mixture model assumes that a data point belongs to one and only one cluster. In order to obtain a model allowing multiple clusters for a data point, the constraint $z_{nk} \in \{0, 1\}$ is relaxed to $z_{nk} \in [0, 1]$, e.g., $\mathbf{z}_n = [0.05, 0.1, 0.75, 0.1]$. Correspondingly, Equation (1.30) is modified to be

$$\begin{aligned} & p(\mathbf{x}_n|\mathbf{z}_n, \mathbf{\Theta}) = \prod_{k=1}^{K} p_k(\mathbf{x}_n|\boldsymbol{\theta}_k)^{z_{nk}}, \\ & z_{nk} \in [0, 1], \\ & \sum_{k=1}^{K} z_{nk} = 1. \end{aligned} \tag{1.32}$$

For the above example with $\mathbf{z}_n = [0.05, 0.1, 0.75, 0.1]$, Equation (1.32) is

$$p(\mathbf{x}_n|\mathbf{z}_n, \boldsymbol{\Theta}) = p_1(\mathbf{x}_n|\boldsymbol{\theta}_1)^{0.05} \cdot p_2(\mathbf{x}_n|\boldsymbol{\theta}_2)^{0.1} \cdot p_3(\mathbf{x}_n|\boldsymbol{\theta}_3)^{0.75} \cdot p_4(\mathbf{x}_n|\boldsymbol{\theta}_4)^{0.1}. \tag{1.33}$$

Comparing Equation (1.31) and (1.33), it can be seen that in Equation (1.31) only one cluster (the 3rd one) participates in the data generation, while in Equation (1.33) all the clusters have their own influence on the data generation.

With the partial membership model defined in Equation (1.32), a Bayesian framework for modeling partial memberships of data points to clusters, Bayesian Partial Membership model (BPM), is defined as,

$$\begin{aligned} p(\boldsymbol{\pi}, s, \mathbf{z}_n, \mathbf{x}_n|\boldsymbol{\alpha}, \lambda, \boldsymbol{\beta}) &= p(\boldsymbol{\pi}|\boldsymbol{\alpha})p(s|\lambda)p(\mathbf{z}_n|\boldsymbol{\pi}s) \\ &\quad \prod_{k=1}^{K} p_k(\mathbf{x}_n|\beta_k)^{z_{nk}}, \\ z_{nk} \in [0,1], &\quad \sum_{k=1}^{K} z_{nk} = 1, \end{aligned} \tag{1.34}$$

where $\boldsymbol{\pi}$ is the cluster mixing proportion, assumed to be distributed according to a Dirichlet distribution with parameter $\boldsymbol{\alpha}$:

$$\boldsymbol{\pi} \sim \text{Dir}(\boldsymbol{\alpha}). \tag{1.35}$$

The scaling factor, s, determines the level of cluster mixing and is distributed according to an exponential distribution with mean $1/\lambda$:

$$s \sim \exp(\lambda). \tag{1.36}$$

$\mathbf{z}_n \sim \text{Dir}(\boldsymbol{\pi}s)$ is the membership vector for data point $\mathbf{x}_n$.

As shown in [29], if each of the mixture components are exponential family distributions of the same type, then $p(\mathbf{x}_n|\mathbf{z}_n, \boldsymbol{\beta}) = \prod_{k=1}^{K} p_k(\mathbf{x}_n|\beta_k)^{z_{nk}}$ with $z_{nk} \in [0, 1]$ and $\sum_{k=1}^{K} z_{nk} = 1$, can be written as:

$$p(\mathbf{x}_n|\mathbf{z}_n, \boldsymbol{\beta}) = \text{Expon}\left(\sum_k z_{nk}\eta_k\right), \tag{1.37}$$

where $\text{Expon}\big(\sum_k z_{nk}\eta_k\big)$ indicates that the data generating distribution for $\mathbf{x}_n$ is of the same exponential family distribution as the original K clusters, but with new natural parameters $\sum_k z_{nk}\eta_k$. The new parameters are a convex combination of the natural parameters, η_k, of the original clusters weighted by z_{nk}. This provides the powerful (and convenient) ability to sample directly from the unique mixture distribution for each data point if the natural parameters of the original clusters and the membership vector for the data point is known. Given this formalization of the partial membership model, a graphical model of BPM is shown in Figure 1.5. The generative process of BPM is described as follows [29]:

1. Draw mixing proportion $\boldsymbol{\pi}$ from a Dirichlet distribution, $\boldsymbol{\pi} \sim \text{Dir}(\boldsymbol{\alpha})$.
2. Draw scaling constant s from an exponential distribution $s \sim \exp(\lambda) = \lambda e^{-\lambda s}$.
3. For each data point $\mathbf{x}_n$
   (a) Draw the membership vector $\mathbf{z}_n$ from a Dirichlet distribution, $\mathbf{z}_n \sim \text{Dir}(s\boldsymbol{\pi})$.
   (b) Draw a data point form $\mathbf{x}_n \sim \text{Expon}(\sum_k z_{nk}\boldsymbol{\eta}_k)$.

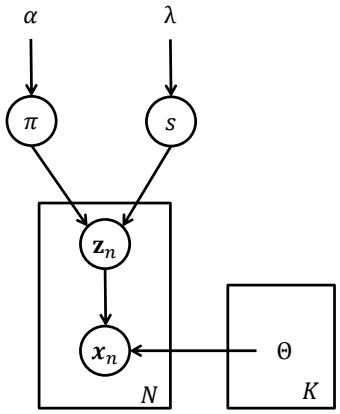


Figure 1.5: BPM

1.2.4 Unifying Model of BFC and BPM

The BFC model is composed of a data likelihood distribution, FDL (Equation (1.26)), a improper prior distribution for the cluster memberships, FCP (Equation (1.24)) and a Gaussian prior distribution on the cluster prototypes (Equation (1.25)). The BPM model is composed of a data likelihood (Equation (1.38)), a membership distribution (Equation (1.39)) and priors on mixing proportion, scaling factor, and clusters (Equation (1.40) - (1.41)). In BPM, the data likelihood is modeled as an exponential family distribution defined as

$$P(\mathbf{X}|\mathbf{Z},\boldsymbol{\beta}) = \prod_{n=1}^{N} \text{Expon}\left(\sum_{k=1}^{K} z_{nk}\beta_k\right), \tag{1.38}$$

and the membership distribution is modeled as a Dirichlet distribution defined as

$$p(\mathbf{Z}|\boldsymbol{\pi}s) = \prod_{n=1}^{N} \frac{\Gamma\left(\sum_{k=1}^{K} s\pi_k\right)}{\prod_{k=1}^{K}\Gamma(s\pi_k)} \prod_{k=1}^{K}(z_{nk})^{(s\pi_k-1)}, \tag{1.39}$$

and the prior on mixing proportion is modeled as a Dirichlet distribution defined as

$$p(\boldsymbol{\pi}|\boldsymbol{\alpha}) = \frac{\Gamma\left(\sum_{k=1}^{K} \alpha_k\right)}{\prod_{k=1}^{K} \Gamma(\alpha_k)} \prod_{k=1}^{K} (\pi_k)^{(\alpha_k - 1)}, \tag{1.40}$$

and the prior on scaling factor is modeled as an exponential distribution defined as

$$p(s|\lambda) = \lambda e^{-\lambda s}, \tag{1.41}$$

and the prior on clusters is modeled as a distribution conjugate to the above exponential family distribution in Equation (1.38).

A table comparison between BFC and BPM is shown in Table 1.1. The major difference between the BPM and BFC models is that the BFC uses both a fixed *fuzzifier* parameter and the membership prior FCP to control the degree of mixing between clusters. In contrast, in the BPM, the degree of mixing between clusters is controlled only through a scaling hyper-parameter, s, found in the prior distribution on partial membership values. The BPM explicitly defines the cluster mixing proportion $\boldsymbol{\pi}$ while the BFC does not. For data generating distribution, coefficients of the convex combination for BFC are powered by *fuzzifer* m while the BPM uses the memberships as coefficients directly.

Table 1.1: Table comparison between BFC, BPM and the unifying model.

	BFC	BPM	Unifying model
fuzzifier	m	$m = 1$	m
Scaling Factor	N/A	$s \sim \exp(\lambda)$	$s \sim \exp(\lambda)$
Mixing Prop.	N/A	$\boldsymbol{\pi} \sim \text{Dir}(\boldsymbol{\alpha})$	$\boldsymbol{\pi} \sim \text{Dir}(\boldsymbol{\alpha})$
Membership	$\mathbf{z} \sim$ FCP	$\mathbf{z} \sim \text{Dir}(\boldsymbol{\pi} s)$	$\mathbf{z} \sim$ Modified FCP
Coeff. of Data Generating Distribution	$\frac{z_{nk}^m}{\sum_k z_{nk}^m}$	$\frac{z_{nk}^m}{\sum_k z_{nk}^m}, m = 1$	$\frac{z_{nk}^m}{\sum_k z_{nk}^m}$

A unifying model combining BFC and BPM is proposed to investigate the effect of *fuzzifier* m and scaling factor s on the membership mixing level. The unifying model is composed of an exponential prior on scaling factor defined as

$$p(s|\lambda) = \lambda e^{-\lambda s}, \tag{1.42}$$

a Dirichlet prior on mixing proportion defined as

$$p(\boldsymbol{\pi}|\boldsymbol{\alpha}) = \frac{\Gamma\left(\sum_{k=1}^K \alpha_k\right)}{\prod_{k=1}^K \Gamma(\alpha_k)} \prod_{k=1}^K (\pi_k)^{(\alpha_k - 1)}, \tag{1.43}$$

a data likelihood in the same form of FDL, defined as

$$p(\mathbf{X}|\mathbf{Z}, \mathbf{Y}) = \prod_{n=1}^N \frac{1}{Z(\mathbf{u}_n, m, \mathbf{Y})} \prod_{k=1}^K \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \mathbf{Q} = z_{nk}^m \mathbf{I}), \tag{1.44}$$

and a prior distribution for the cluster membership defined as

$$\tilde{p}(\mathbf{Z}|\mathbf{Y}) = \prod_{n=1}^{N} Z(\mathbf{u}_n, m, \mathbf{Y}) \left(\prod_{k=1}^{K} z_{nk}^{-mp/2} \right) \text{Dir}(\mathbf{z}_n | s\boldsymbol{\pi}), \tag{1.45}$$

which modifies the FCP in BFC by replacing the Dirichlet parameter with $s\boldsymbol{\pi}$. The joint likelihood of the unifying model can be computed as

$$\begin{aligned}
p(\mathbf{X}, \mathbf{Z}, \boldsymbol{\pi}, s|\mathbf{Y}) &= p(\mathbf{X}|\mathbf{Z}, \mathbf{Y}) \tilde{p}(\mathbf{Z}|\mathbf{Y}) p(\boldsymbol{\pi}) p(s) \\
&= \prod_{n=1}^{N} \frac{1}{Z(\mathbf{u}_n, m, \mathbf{Y})} \prod_{k=1}^{K} \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \mathbf{Q} = z_{nk}^{m} \mathbf{I}) \\
&\quad \prod_{n=1}^{N} Z(\mathbf{u}_n, m, \mathbf{Y}) \left(\prod_{k=1}^{K} z_{nk}^{-mp/2} \right) \frac{\Gamma\left(\sum_{k=1}^{K} s\pi_k \right)}{\prod_{k=1}^{K} \Gamma(s\pi_k)} \prod_{k=1}^{K} (z_{nk})^{(s\pi_k - 1)} \\
&\quad \frac{\Gamma\left(\sum_{k=1}^{K} \alpha_k \right)}{\prod_{k=1}^{K} \Gamma(\alpha_k)} \prod_{k=1}^{K} (\pi_k)^{(\alpha_k - 1)} \lambda e^{-\lambda s} \\
&= \prod_{n=1}^{N} \prod_{k=1}^{K} \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \mathbf{Q} = z_{nk}^{m} \mathbf{I}) z_{nk}^{-mp/2} \\
&\quad \frac{\Gamma\left(\sum_{k=1}^{K} s\pi_k \right)}{\prod_{k=1}^{K} \Gamma(s\pi_k)} \prod_{k=1}^{K} (z_{nk})^{(s\pi_k - 1)} \frac{\Gamma\left(\sum_{k=1}^{K} \alpha_k \right)}{\prod_{k=1}^{K} \Gamma(\alpha_k)} \prod_{k=1}^{K} (\pi_k)^{(\alpha_k - 1)} \lambda e^{-\lambda s} \\
&= \prod_{n=1}^{N} \prod_{k=1}^{K} (2\pi)^{-p/2} \exp\left\{ -\frac{1}{2} z_{nk}^{m} (\mathbf{x}_n - \boldsymbol{\mu}_k)^T (\mathbf{x}_n - \boldsymbol{\mu}_k) \right\} \\
&\quad \frac{\Gamma\left(\sum_{k=1}^{K} s\pi_k \right)}{\prod_{k=1}^{K} \Gamma(s\pi_k)} \prod_{k=1}^{K} (z_{nk})^{(s\pi_k - 1)} \frac{\Gamma\left(\sum_{k=1}^{K} \alpha_k \right)}{\prod_{k=1}^{K} \Gamma(\alpha_k)} \prod_{k=1}^{K} (\pi_k)^{(\alpha_k - 1)} \lambda e^{-\lambda s}.
\end{aligned} \tag{1.46}$$

The log of this joint likelihood is defined as

$$\begin{aligned}
\ln p(\mathbf{X}, \mathbf{Z}, \mathbf{Y}, s, \boldsymbol{\pi}) &= \text{const} + \sum_{n=1}^{N}\sum_{k=1}^{K}\left\{-\frac{1}{2} z_{nk}^{m}(\mathbf{x}_n - \boldsymbol{\mu}_k)^T(\mathbf{x}_n - \boldsymbol{\mu}_k)\right\} \\
&+ \sum_{n=1}^{N}\left\{\ln\Gamma\left(\sum_{k=1}^{K} s\pi_k\right) - \sum_{k=1}^{K}\ln\Gamma(s\pi_k) + \sum_{k=1}^{K}(s\pi_k - 1)\ln z_{nk}\right\} \\
&+ \ln\Gamma\left(\sum_{k=1}^{K}\alpha_k\right) - \sum_{k=1}^{K}\ln\Gamma(\alpha_k) + \sum_{k=1}^{K}(\alpha_k - 1)\ln\pi_k \\
&+ \ln\lambda - \lambda s \qquad (1.47)
\end{aligned}$$

An experiment was conducted to show the effect of m and s on the membership mixing level in the unifying model. The two Gaussian clusters are set to be $\mathcal{N}(0,1)$ and $\mathcal{N}(1,1)$. A set of input data points $\mathbf{X}$ are generated in range $[-0.5, 1.5]$ with increment 0.05, and a set of memberships $\mathbf{Z}$ are generated in range $[0,1]$ with increment 0.05. The *fuzzifier* m is varied to be -1, -0.5, 0, 0.5, 1, 2, 5, and 10. The scaling factor s is varied to be 0.1, 0.5, 1, 2, and 5. The parameter λ is set to be 1. The cluster mixing proportion $\boldsymbol{\pi}$ is fixed to be $[0.5, 0.5]$. The parameter $\boldsymbol{\alpha}$ is set to be $[1,1]$. The log of joint likelihood in the unifying model is shown in Figure 1.6. X axis denotes the membership in the first cluster ranging from 0 to 1 and Y axis denotes the input data point x ranging from -0.5 to 1.5. As the cluster mixing proportion $\boldsymbol{\pi}$ is fixed, the scaling factor s directly determines the prior on memberships. As shown in each column where *fuzzifier* m is fixed, as s increases, mixing memberships get higher log likelihood. Memberships with high log likelihood (bright yellow region) are gradually moved from the left and right margins (crisp membership) to the horizontal center (mixing membership). The *fuzzifier* m determines the vertical position of the high log likelihood region (which data points have high log likelihood). As shown in each row, with m increases, data points with high log likelihood are gradually

expanded to the top and bottom margins. The scaling factor s impacts the cluster mixing level by controlling the prior knowledge on memberships and the *fuzzifier* m by controlling the percentage of mixing data points.

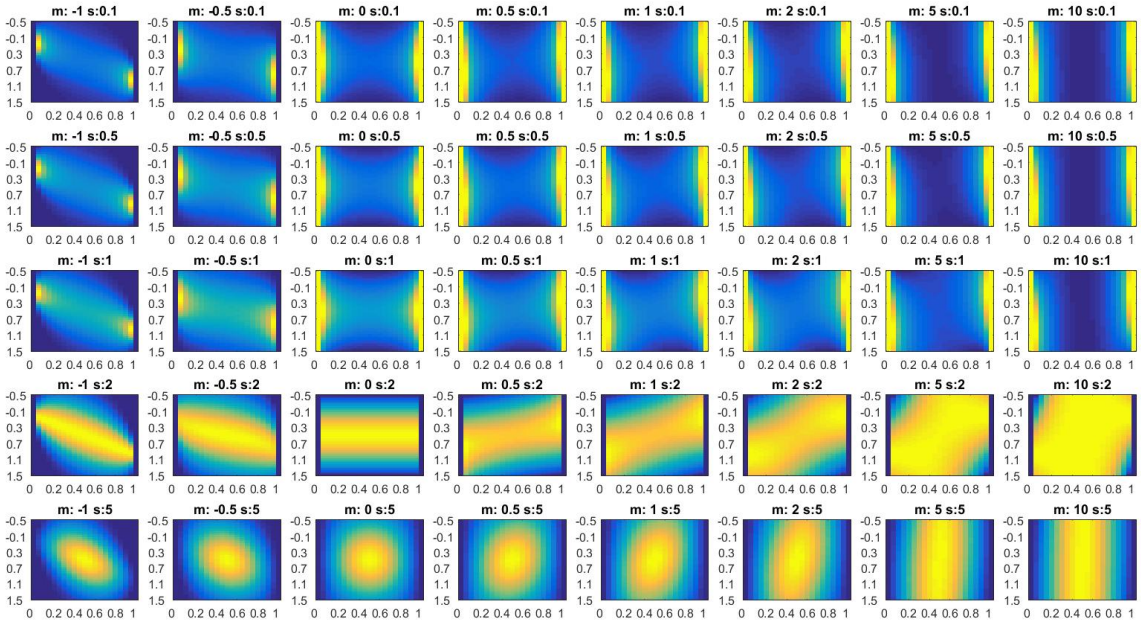


Figure 1.6: Log likelihood of the unifying model with different m and s. X axis denotes the membership in the first cluster varied from 0 to 1. Y axis denotes the input data point x varied from -0.5 to 1.5.

1.3 Cluster Representation

In this work, the form of the distribution for each cluster is assumed to be in the exponential family. This section focuses on Gaussian distribution and multinomial distribution

specifically. An exponential family distribution can be written as:

$$\begin{aligned} p(\mathbf{x}|\boldsymbol{\eta}) &= h(\mathbf{x})g(\boldsymbol{\eta})\text{exp}\left\{\boldsymbol{\eta}^T\mathbf{s}(\mathbf{x})\right\} \\ &= \text{Expon}(\mathbf{x}|\boldsymbol{\eta}), \end{aligned} \tag{1.48}$$

where $\boldsymbol{s}(\mathbf{x})$ is known as sufficient statistics (a statistic is a function of data), $\boldsymbol{\eta}$ are natural parameters, $h(\boldsymbol{x})$ is the underlying measure, also depending on the data, and $g(\boldsymbol{\eta})$ is the normalizer, ensuring the probability sums/integrates to 1. For Gaussian distribution, $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$, the natural parameters $\boldsymbol{\eta} = (\boldsymbol{\Sigma}^{-1}, \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu})$, and for multinomial(categorical) distribution, $p(\mathbf{x}|\boldsymbol{\beta}) = \prod_{m=1}^{M}\beta_m^{x_m}$, the natural parameters $\eta_m = \ln\left(\frac{\beta_m}{\beta_M}\right), m = 1, \cdots, M$.

For the k-th cluster with natural parameters $\boldsymbol{\eta}_k$, its distribution can be rewritten as

$$\begin{aligned} p_k(\mathbf{x}_n|\boldsymbol{\eta}_k) &= h(\mathbf{x}_n)g(\boldsymbol{\eta}_k)\text{exp}\left\{\boldsymbol{\eta}_k{}^T\mathbf{s}(\mathbf{x}_n)\right\} \\ &= \text{Expon}(\mathbf{x}_n|\boldsymbol{\eta}_k). \end{aligned} \tag{1.49}$$

Plugging Equation (1.49) into Equation (1.32), Equation (1.32) can be rewritten as an exponential family distribution

$$\begin{aligned} p(\mathbf{x}_n|\mathbf{z}_n, \boldsymbol{\Theta}) &= \prod_{k=1}^{K} p_k(\mathbf{x}_n|\boldsymbol{\theta}_k)^{z_{nk}} \\ &= \prod_{k=1}^{K} \text{Expon}(\mathbf{x}_n|\boldsymbol{\eta}_k)^{z_{nk}} \\ &= \text{Expon}\left(\mathbf{x}_n \middle| \sum_{k=1}^{K} z_{nk}\boldsymbol{\eta}_k\right), \end{aligned} \tag{1.50}$$

where $p(\mathbf{x}_n|\mathbf{z}_n, \boldsymbol{\Theta})$ has the same distribution form as the original clusters, and its natural parameters is a convex combination of the natural parameters of the original clusters, $\boldsymbol{\eta}_k$,

weighted by the partial membership z_{nk}.

1.3.1 Cluster as a Gaussian Distribution

Each cluster can be represented with a Gaussian (continuous) distribution. The cluster parameter $\boldsymbol{\theta}_k = (\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ can be converted to the natural parameters $\boldsymbol{\eta}_k = (\boldsymbol{\Lambda}_k, \boldsymbol{\Lambda}_k\boldsymbol{\mu}_k)$, where $\boldsymbol{\Lambda}_k = \boldsymbol{\Sigma}_k^{-1}$ is called the precision matrix. For Gaussian clusters, Equation (1.50) becomes

$$\begin{aligned} p(\mathbf{x}_n|\mathbf{z}_n, \boldsymbol{\Theta}) &= \text{Expon}(\mathbf{x}_n|\boldsymbol{\Lambda}'_n, \boldsymbol{\Lambda}'_n\boldsymbol{\mu}'_n), \\ &= \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}'_n, \boldsymbol{\Sigma}'_n) \end{aligned} \tag{1.51}$$

where

$$\boldsymbol{\Lambda}'_n = \sum_{k=1}^{K} z_{nk}\boldsymbol{\Lambda}_k \qquad \rightarrow \qquad \boldsymbol{\Sigma}'_n = \left(\sum_{k=1}^{K} z_{nk}{\boldsymbol{\Sigma}_k}^{-1}\right)^{-1} \tag{1.52}$$

$$\boldsymbol{\Lambda}'_n\boldsymbol{\mu}'_n = \sum_{k=1}^{K} z_{nk}\boldsymbol{\Lambda}_k\boldsymbol{\mu}_k \qquad \rightarrow \qquad \boldsymbol{\mu}'_n = \boldsymbol{\Sigma}'_n \sum_{k=1}^{K} z_{nk}{\boldsymbol{\Sigma}_k}^{-1}\boldsymbol{\mu}_k. \tag{1.53}$$

For Gaussian clusters, a data point $\mathbf{x}_n$ is drawn from a new Gaussian distribution with natural parameters as a convex combination of the natural parameters of the original clusters, $\boldsymbol{\eta}_k = (\boldsymbol{\Lambda}_k, \boldsymbol{\Lambda}_k\boldsymbol{\mu}_k)$, weighted by the partial membership z_{nk}. Different membership values yield different Gaussian distributions.

1.3.2 Cluster as a Categorical Distribution

As another example, each cluster can be represented with a Categorical (discrete) distribution. The cluster parameter $\boldsymbol{\theta}_k = \{\beta_{k1}, \beta_{k2}, ..., \beta_{kM}\}$ can be converted to the natural parameters $\boldsymbol{\eta}_k = \{\eta_{k1}, \eta_{k2}, ..., \eta_{kM}\}$ using the following equations,

$$\eta_{km} = \ln\left(\frac{\beta_{km}}{\beta_{kM}}\right), \qquad \beta_{km} = \frac{\exp\left(\eta_{km}\right)}{\sum_{j=1}^{M}\exp\left(\eta_{kj}\right)}. \tag{1.54}$$

For categorical distributed clusters, Equation (1.50) becomes

$$\begin{aligned} p(\mathbf{x}_n|\mathbf{z}_n, \boldsymbol{\Theta}) &= \text{categorical}(\boldsymbol{\eta}'_n) \\ &= \text{categorical}(\boldsymbol{\beta}'_n) \end{aligned} \tag{1.55}$$

where

$$\eta'_{nm} = \sum_{k=1}^{K} z_{nk}\eta_{km} = \sum_{k=1}^{K} z_{nk}\ln\left(\frac{\beta_{km}}{\beta_{kM}}\right) = \sum_{k=1}^{K}\ln\left(\frac{\beta_{km}}{\beta_{kM}}\right)^{z_{nk}} = \ln\left(\prod_{k=1}^{K}\left(\frac{\beta_{km}}{\beta_{kM}}\right)^{z_{nk}}\right), \tag{1.56}$$

$$\beta'_{nm} = \frac{\exp\left(\eta'_{nm}\right)}{\sum_{j=1}^{M}\exp\left(\eta'_{nj}\right)} = \frac{\prod_{k=1}^{K}(\beta_{km})^{z_{nk}}}{\sum_{j=1}^{F}\left(\prod_{k=1}^{K}\left(\beta_{kj}\right)^{z_{nk}}\right)}. \tag{1.57}$$

For categorical clusters, the new natural parameter η'_{nm} is a convex combination of the natural parameter of the original clusters, η_{km}, weighted by the partial membership z_{nk}; the new β'_{nm} can be considered as a normalized weighted product of the mth parameter, β_{km}, of all the clusters.

Chapter 2

Proposed Algorithm

The standard LDA model assumes that each word comes from one and only one topic. This is generally true for text documents, since a text word is the smallest meaningful unit that cannot be broken down any further. For image documents, however, a visual word may belongs to multiple topics. For example, consider the photograph in Figure 2.1a where the gradually thinning fog blurs the boundary between the foggy sky and the mountain, a sharp boundary between the "fog" and "mountain" topics does not exist. Similarly, in Figure 2.1b consider the gradually fading sunlight or in Figure 2.1c consider the gradually vanishing sand ripples shown in the Synthetic Aperture SONAR image of the sea floor. In both of these cases, sharp boundaries between the "sun" and "sky" topics or "sand ripple" and "flat sand"topics do not exist. To address this, we propose to allow visual words in regions of transition to have partial memberships in multiple topics, i.e., in both the "fog" and "mountain" topics. A partial membership model for Latent Dirichlet Allocation is, thus far, an untouched topic in the literature. In this work, the standard LDA is generalized to allow for partial memberships. A Partial Membership Latent Dirichlet Allocation model

in which the words can have partial membership in all available topics is proposed. A generative process for PM-LDA model which contains extra steps to generate the partial membership is established, and the inference and parameter estimation methods based on Gibbs sampler are developed.

(a) (b) (c)

Figure 2.1: Examples of images with gradual transition region. (a) A foggy mountain photograph with a gradual transition from fog to mountain [**?**]. (b) A sunset photograph with a gradual transition from sun to sky [**?**]. (c) A SONAR image with gradually vanishing sand ripples.

2.1 Partial Membership Latent Dirichlet Allocation

As discussed in [29, 44], in the BPM, data points are organized at only one level, where each data point is indexed by its corresponding component distribution. In the proposed model, PM-LDA, and in the original LDA model data is organized at two levels: the word level and the document level. This two-level organization is illustrated in Figure 2.3 and 2.4. In the proposed PM-LDA model, the random variable associated with a data point is assumed to be distributed according to multiple topics with a continuous partial membership in each

topic. Specifically, the PM-LDA model is

$$p(\boldsymbol{\pi}^d, s^d, \mathbf{z}_n^d, \mathbf{x}_n^d | \boldsymbol{\alpha}, \lambda, \boldsymbol{\beta}) = p(\boldsymbol{\pi}^d|\boldsymbol{\alpha}) p(s^d|\lambda) p(\mathbf{z}_n^d|\boldsymbol{\pi}^d s^d) \prod_{k=1}^{K} p_k(\mathbf{x}_n^d|\beta_k)^{z_{nk}^d} \quad (2.1)$$

where the document index d is moved to the superscript, $\mathbf{x}_n^d$ is the nth word in document d, $\mathbf{z}_n^d$ is the partial membership vector of $\mathbf{x}_n^d$, $\boldsymbol{\pi}^d \sim \text{Dir}(\boldsymbol{\alpha})$ and $s^d \sim \exp(\lambda)$ are the topic proportion and the level of topic mixing in document d, respectively. The parameter $\boldsymbol{\alpha}$ gives the topic composition across a document. For example, in a synthetic aperture SONAR image depicting an area of sea floor (e.g., Figure 2.1c), the image may be composed of 40% of the "sand ripple" topic and 60% of the "flat sand" topic. The parameter λ controls how similar the partial membership vector of each word is expected to be to the topic distribution of the document. For example, a small λ would correspond to most words in an document to have partial membership vectors very close to $\boldsymbol{\pi}^d$. In an image segmentation application, this generally corresponds to large transition regions between topics (e.g,, the transition from "flat sand" to "sand ripple" is very slow and comprises the majority of the image). For a large λ, the partial membership vectors for each word can vary significantly from the document mixing proportions. In general, this corresponds to very narrow (tending towards crisp) transition regions in an image segmentation application (e.g., the SAS image may have 39% of the visual words as pure "sand ripple", 59% as pure "flat sand", and only 2% mixed). The vector $\mathbf{z}_n^d$ represents the partial memberships of data point $\mathbf{x}_n^d$ in each of the K topics. The vector $\mathbf{z}_n^d$ is distributed according to a Dirichlet distribution,

$$\mathbf{z}_n^d \sim \text{Dir}(\boldsymbol{\pi}^d s^d). \quad (2.2)$$

If each component distribution is assumed to be of the exponential family, $p_k(\cdot|\beta_k) = \text{Expon}(\eta_k)$, then using the result in (1.37), $p(\mathbf{x}_n^d|\mathbf{z}_n^d, \boldsymbol{\beta}) = \text{Expon}(\sum_k z_{nk}^d \eta_k)$.

The corresponding graphical model of PM-LDA is shown in Figure 2.4 and the generative procedure of PM-LDA is described as follows.

1. For each document $\mathbf{X}^d$, draw topic proportions $\boldsymbol{\pi}^d \sim \text{Dir}(\boldsymbol{\alpha})$
2. Draw scaling factor s^d from an exponential distribution $s^d \sim \exp(\lambda) = \lambda e^{-\lambda s}$.
3. For each word $\mathbf{x}_n^d$
   (a) Draw the membership vector $\mathbf{z}_n^d \sim \text{Dir}(\boldsymbol{\pi}^d s^d)$.
   (b) Draw word $\mathbf{x}_n^d \sim \text{Expon}(\sum_k z_{nk}^d \eta_k)$

In PM-LDA, the membership $\mathbf{z}_n^d$ is drawn from a Dirichlet distribution which is in contrast to a multinomial distribution as used in LDA. With the infinite number of possible values for $\mathbf{z}_n^d$, the word generating distributions in PM-LDA are expanded from only K generating distributions (as in LDA) to infinitely many. Figure 2.5 illustrates this using two Gaussian topic distributions, where the membership value to one topic is varied from 0 to 1 with an increment 0.1. The two original topics are shown as the Gaussian distributions at either end. In LDA, words are generated from only the two original topic distributions. In PM-LDA, words can be generated from any of the topic distributions including and between the original topic distributions. As the scaling factor $s \to 0$, the PM-LDA model will degrade to the LDA model.

Given the hyperparameters $\boldsymbol{\Psi} = \{\boldsymbol{\alpha}, \lambda, \boldsymbol{\beta}\}$, the full PM-LDA model over all words in a document can be written as the joint probability of the topic proportions $\boldsymbol{\pi}^d$, scaling factor

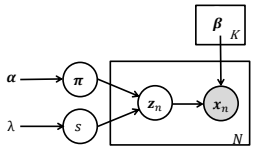


Figure 2.2: BPM

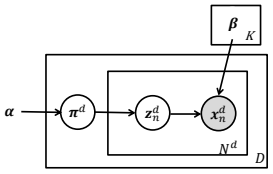


Figure 2.3: LDA

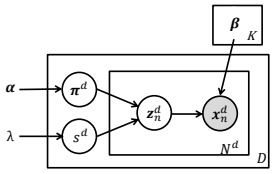
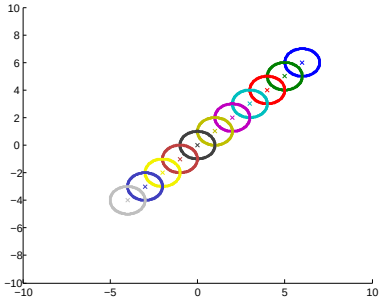
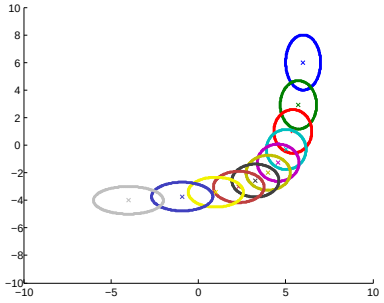


Figure 2.4: PM-LDA



(a)



(b)

Figure 2.5: Word generating distributions in partial membership model. In (a), the two Gaussian topics have means $\mu_1 = [-4 \quad -4]; \mu_2 = [6 \quad 6]$ and covariances $\Sigma_1 = \Sigma_2 = \mathbf{I}$. In (b), the two Gaussian topics have means $\mu_1 = [-4 \quad -4]; \mu_2 = [6 \quad 6]$ with covariances of $\Sigma_1 = [4 \quad 0; 0 \quad 1], \Sigma_2 = [1 \quad 0; 0 \quad 4]$. This figure is adapted from [29].

s^d, partial membership vectors $\mathbf{Z}^d = \{\mathbf{z}_n^d\}_{n=1}^{N^d}$ and a set of N^d words $\mathbf{X}^d$, defined as:

$$\begin{aligned} & p(\boldsymbol{\pi}^d, s^d, \mathbf{Z}^d, \mathbf{X}^d | \boldsymbol{\alpha}, \lambda, \boldsymbol{\beta}) \\ = \; & p(\boldsymbol{\pi}^d|\boldsymbol{\alpha}) p(s^d|\lambda) \prod_{n=1}^{N^d} p(\mathbf{x}_n^d|\mathbf{z}_n^d, \boldsymbol{\beta}) p(\mathbf{z}_n^d|\boldsymbol{\pi}^d s^d). \end{aligned} \tag{2.3}$$

When considering the specific distribution forms chosen in PM-LDA model that

$$\begin{aligned} p(\boldsymbol{\pi}^d|\boldsymbol{\alpha}) &= \frac{\Gamma\left(\sum_{k=1}^K \alpha_k\right)}{\prod_{k=1}^K \Gamma(\alpha_k)} \prod_{k=1}^K (\pi_k^d)^{(\alpha_k - 1)} \\ p(s^d|\lambda) &= \lambda e^{-\lambda s^d} \\ p(\mathbf{z}_n^d|\boldsymbol{\pi}^d s^d) &= \frac{\Gamma\left(\sum_{k=1}^K s^d \pi_k^d\right)}{\prod_{k=1}^K \Gamma(s^d \pi_k^d)} \prod_{k=1}^K (z_{nk}^d)^{(s^d \pi_k^d - 1)}, \end{aligned}$$

The joint probability in Equation (2.3) is

$$
\begin{aligned}
& p(\boldsymbol{\pi}^d, s^d, \mathbf{Z}^d, \mathbf{X}^d | \boldsymbol{\alpha}, \lambda, \boldsymbol{\beta}) \\
= \; & p(\boldsymbol{\pi}^d | \boldsymbol{\alpha}) p(s^d | \lambda) \prod_{n=1}^{N^d} p(\mathbf{x}_n^d | \mathbf{z}_n^d, \boldsymbol{\beta}) p(\mathbf{z}_n^d | \boldsymbol{\pi}^d s^d). \\
= \; & \left(\frac{\Gamma\left(\sum_{k=1}^K \alpha_k\right)}{\prod_{k=1}^K \Gamma(\alpha_k)} \prod_{k=1}^K (\pi_k^d)^{(\alpha_k - 1)} \right) \left(\lambda e^{-\lambda s^d} \right) \\
& \prod_{n=1}^{N^d} \left(p(\mathbf{x}_n^d | \mathbf{z}_n^d, \boldsymbol{\beta}) \frac{\Gamma\left(\sum_{k=1}^K s^d \pi_k^d\right)}{\prod_{k=1}^K \Gamma(s^d \pi_k^d)} \prod_{k=1}^K (z_{nk}^d)^{(s^d \pi_k^d - 1)} \right).
\end{aligned}
\tag{2.4}
$$

The log of this joint distribution in Equation (2.4), is:

$$
\begin{aligned}
& \ln(p(\boldsymbol{\pi}^d, s^d, \mathbf{Z}^d, \mathbf{X}^d | \boldsymbol{\alpha}, \lambda, \boldsymbol{\beta})) = \\
& \ln p(\boldsymbol{\pi}^d | \boldsymbol{\alpha}) + \ln p(s^d | \lambda) \\
+ \; & \sum_{n=1}^{N^d} \left\{ \ln p(\mathbf{x}_n^d | \mathbf{z}_n^d, \boldsymbol{\beta}) + \ln p(\mathbf{z}_n^d | \boldsymbol{\pi}^d s^d) \right\} \\
= \; & \ln \Gamma\left(\sum_{k=1}^K \alpha_k \right) - \sum_{k=1}^K \ln \Gamma(\alpha_k) \\
+ \; & \sum_{k=1}^K (\alpha_k - 1) \ln \pi_k^d + \ln \lambda - \lambda s^d \\
+ \; & \sum_{n=1}^{N^d} \ln p(\mathbf{x}_n^d | \mathbf{z}_n^d, \boldsymbol{\beta}) + \sum_{n=1}^{N^d} \left\{ \ln \Gamma\left(\sum_{k=1}^K s^d \pi_k^d \right) \right. \\
- \; & \left. \sum_{k=1}^K \ln \Gamma(s^d \pi_k^d) + \sum_{k=1}^K (s^d \pi_k^d - 1) \ln z_{nk}^d \right\}
\end{aligned}
\tag{2.5}
$$

The BPM model and PM-LDA model originate from the mixture model and the mixed membership model, respectively, and both extend the original models by introducing the partial membership. The PM-LDA model can be considered as an extension of the BPM model in the same way as the mixed membership model is an extension of the mixture model. The mixed membership model is a mixture of mixture model where the data is organized in two levels [29, 45]. Each "data point" is a collection of data, and each collection can belong to multiple groups [46]. In the PM-LDA model, the data is organized at the level of words and then documents. Each document is a collection of words, and each document can belong to multiple topics. The latent variables in the proposed model include the document-specific topic proportion and the partial memberships. The BMP model indexes the data directly and the latent variable represents the partial memberships only [44].

2.2 Inference using Gibbs Sampler for One Document

The goal of inference given one document is to predict the topic proportion $\boldsymbol{\pi}$, the scaling factor s, and the memberships $\mathbf{Z}$, given the document $\mathbf{X}$, the topic proportion prior $\boldsymbol{\alpha}$, the scaling factor prior λ, and the cluster parameters $\boldsymbol{\Theta}$. Let $\boldsymbol{\Omega} = \{\boldsymbol{\pi}, s, \mathbf{Z}\}$ denote the variables to predict. The inference amounts to finding $\boldsymbol{\Omega}$ that maximizes the following posterior distribution

$$p(\boldsymbol{\Omega}|\mathbf{X}, \boldsymbol{\Psi}) = \frac{p(\boldsymbol{\Omega}, \mathbf{X}|\boldsymbol{\Psi})}{p(\mathbf{X}|\boldsymbol{\Psi})}, \tag{2.6}$$

which is intractable to compute since the denominator, $p(\mathbf{X}|\boldsymbol{\Psi})$, cannot be computed exactly. In this work, a Metropolis within Gibbs sampler is employed to perform the MAP inference which can generate samples from the posterior distribution in (2.5) [43], where the topic proportion $\boldsymbol{\pi}$, scaling factor s, and memberships $\mathbf{Z}$ are sampled alternately. Algo-

rithm 2 outlines the procedure for this method.

Algorithm 2 Gibbs Sampling Method for Inference

Input: A document $\mathbf{X}$, K topics, and the number of sampling iterations T
Output: Collection of all samples: $\boldsymbol{\pi}^{(t)}, s^{(t)}, \mathbf{Z}^{(t)}$.

1: **for** $t = 1 : T$ **do**
2: Sample $\boldsymbol{\pi}$ using (2.14) to (2.15)
3: Sample s using (2.16) to (2.17)
4: **for** $n = 1 : N$ **do**
5: Sample $\mathbf{z}_n$ using (2.18) to (2.19)
6: **end for**
7: **end for**

Step 2. Sample the topic proportion vector $\boldsymbol{\pi}$ A proposed topic proportion $\boldsymbol{\pi}^{\dagger}$ for the document is sampled from the prior Dirichlet proposal distribution

$$\boldsymbol{\pi}^{\dagger} \sim \text{Dir}(\boldsymbol{\alpha}), \tag{2.7}$$

which will be accepted with the probability

$$a_{\boldsymbol{\pi}} = \min\left\{1, \frac{p(\boldsymbol{\pi}^{\dagger}, s^{(t-1)}, \mathbf{Z}^{(t-1)}, \mathbf{X}|\boldsymbol{\Psi})p(\boldsymbol{\pi}^{(t-1)}|\boldsymbol{\alpha})}{p(\boldsymbol{\pi}^{(t-1)}, s^{(t-1)}, \mathbf{Z}^{(t-1)}, \mathbf{X}|\boldsymbol{\Psi})p(\boldsymbol{\pi}^{\dagger}|\boldsymbol{\alpha})}\right\}. \tag{2.8}$$

Step 3. Sample the scaling factor s: A proposed scaling factor $s^{\dagger}$ for the document is sampled from the prior exponential distribution

$$s^{\dagger} \sim \exp(\lambda), \tag{2.9}$$

which will be accepted with the probability

$$a_s = \min\left\{1, \frac{p(\boldsymbol{\pi}^{(t)}, s^{\dagger}, \mathbf{Z}^{(t-1)}, \mathbf{X}|\boldsymbol{\Psi})p(s^{(t-1)}|\lambda)}{p(\boldsymbol{\pi}^{(t)}, s^{(t-1)}, \mathbf{Z}^{(t-1)}, \mathbf{X}|\boldsymbol{\Psi})p(s^{\dagger}|\lambda)}\right\}. \tag{2.10}$$

Step 5. Sample the membership vector $\mathbf{z}_n$ of the nth word: A new membership vector candidate $\mathbf{z}_n^{\dagger}$ is sampled from a uniform symmetric Dirichlet proposal distribution for simplicity

$$\mathbf{z}_n^{\dagger} \sim \text{Dir}(\mathbf{1}_K), \tag{2.11}$$

which will be accepted with the probability

$$a_{\mathbf{z}} = \min\left\{1, \frac{p(\boldsymbol{\pi}^{(t)}, s^{(t)}, \mathbf{z}_n^{\dagger}, \mathbf{x}_n|\boldsymbol{\Psi})}{p(\boldsymbol{\pi}^{(t)}, s^{(t)}, \mathbf{z}_n^{(t-1)}, \mathbf{x}_n|\boldsymbol{\Psi})}\right\}, \tag{2.12}$$

where a Hastings correction is not needed because the proposal distribution is symmetric.

The proposed Metropolis within Gibbs scheme will return the full distribution of parameter values given the desired posterior. In this word, the full MAP sample (i.e., the sample with the largest log likelihood value) is used as the final estimate for the desired parameters, $\{\boldsymbol{\pi}^*, s^*, \mathbf{Z}^*\}$. Given multiple documents, the inference process should be applied to each document sequentially and independently.

2.3 Parameter Estimation using Gibbs Sampler

The goal of parameter estimation is to maximize the following posterior distribution,

$$p(\boldsymbol{\Pi}, \mathbf{S}, \mathbf{M}, \boldsymbol{\beta}|\mathbf{D}, \boldsymbol{\alpha}, \lambda) \propto p(\boldsymbol{\Pi}, \mathbf{S}, \mathbf{M}, \mathbf{D}|\boldsymbol{\alpha}, \lambda, \boldsymbol{\beta}), \tag{2.13}$$

where $\mathbf{D} = \left\{\mathbf{X}^1, \mathbf{X}^2, ..., \mathbf{X}^D\right\}$ includes all training documents and $\mathbf{\Pi}, \mathbf{S}, \mathbf{M}$ include all of the topic proportions, scaling factors and membership vectors, respectively.

The sampling method cycles between the following two stages until convergence:

1. For each document, $\mathbf{X}^d$, sample $\boldsymbol{\pi}^d, s^d, \mathbf{Z}^d$ alternately using a Gibbs sampler, assuming the topic component parameters $\boldsymbol{\beta}$ are fixed.

2. Sample topic component parameters $\boldsymbol{\beta}$ using all the documents, assuming $\mathbf{\Pi}, \mathbf{S}, \mathbf{M}$ are fixed.

The parameter estimation procedure is described in Algorithm 3. Note that to simplify the expressions used, in the following discussion, the document index d is ignored in the topic proportion $\boldsymbol{\pi}^d$, scaling factor s^d, the membership vector $\mathbf{z}_n^d$, and document $\mathbf{X}^d$. Details of Algorithm 3 are described as follows.

Algorithm 3 Gibbs Sampling Method for Parameter Estimation

Input: A corpus $\mathbf{D}$, the number of topics K, and the number of sampling iterations T
Output: Collection of all samples: $\mathbf{\Pi}^{(t)}, \mathbf{S}^{(t)}, \mathbf{M}^{(t)}, \boldsymbol{\beta}^{(t)}$.

for $t = 1 : T$ **do**
  for $d = 1 : D$ **do**
    Sample $\boldsymbol{\pi}^d$ using (2.14) to (2.15)
    Sample s^d using (2.16) to (2.17)
    for $n = 1 : N^d$ **do**
      Sample $\mathbf{z}_n^d$ using (2.18) to (2.19)
    end for
  end for
  for $k = 1 : K$ **do**
    Sample the kth topic
  end for
end for

Step 3. Sample the topic proportion vector $\boldsymbol{\pi}$ of the dth document: A proposed topic proportion $\boldsymbol{\pi}^{\dagger}$ for the dth document is sampled from the prior Dirichlet proposal distribution

$$\boldsymbol{\pi}^{\dagger} \sim \text{Dir}(\boldsymbol{\alpha}), \tag{2.14}$$

which will be accepted with the probability

$$a_{\boldsymbol{\pi}} = \min\left\{1, \frac{p(\boldsymbol{\pi}^{\dagger}, s^{(t-1)}, \mathbf{Z}^{(t-1)}, \mathbf{X}|\boldsymbol{\Psi})p(\boldsymbol{\pi}^{(t-1)}|\boldsymbol{\alpha})}{p(\boldsymbol{\pi}^{(t-1)}, s^{(t-1)}, \mathbf{Z}^{(t-1)}, \mathbf{X}|\boldsymbol{\Psi})p(\boldsymbol{\pi}^{\dagger}|\boldsymbol{\alpha})}\right\}. \tag{2.15}$$

Step 4. Sample the scaling factor s of the dth document: A proposed scaling factor $s^{\dagger}$ for the dth document is sampled from the prior exponential distribution

$$s^{\dagger} \sim \exp(\lambda), \tag{2.16}$$

which will be accepted with the probability

$$a_s = \min\left\{1, \frac{p(\boldsymbol{\pi}^{(t)}, s^{\dagger}, \mathbf{Z}^{(t-1)}, \mathbf{X}|\boldsymbol{\Psi})p(s^{(t-1)}|\lambda)}{p(\boldsymbol{\pi}^{(t)}, s^{(t-1)}, \mathbf{Z}^{(t-1)}, \mathbf{X}|\boldsymbol{\Psi})p(s^{\dagger}|\lambda)}\right\}. \tag{2.17}$$

Step 6. Sample the membership vector $\mathbf{z}_n$ of the nth word in the dth document: A new membership vector candidate $\mathbf{z}_n^{\dagger}$ is sampled from a uniform symmetric Dirichlet proposal distribution for simplicity

$$\mathbf{z}_n^{\dagger} \sim \text{Dir}(\mathbf{1}_K), \tag{2.18}$$

which will be accepted with the probability

$$a_{\mathbf{z}} = \min\left\{1, \frac{p(\boldsymbol{\pi}^{(t)}, s^{(t)}, \mathbf{z}_n^{\dagger}, \mathbf{x}_n|\boldsymbol{\Psi})}{p(\boldsymbol{\pi}^{(t)}, s^{(t)}, \mathbf{z}_n^{(t-1)}, \mathbf{x}_n|\boldsymbol{\Psi})}\right\}, \tag{2.19}$$

where a Hastings correction is not needed because the proposal distribution is symmetric.

The proposed Metropolis within Gibbs scheme will return the full distribution of parameter values given the desired posterior. Our method uses the full MAP sample (i.e., the sample with the largest log likelihood value) as the final estimate for the desired parameters, $\{\boldsymbol{\Pi}^*, \mathbf{S}^*, \mathbf{M}^*, \boldsymbol{\beta}^*\}$.

2.3.1 Topic as Gaussian Distribution

In the current implementation, the topic component distributions are considered to be Gaussian distributions with different means but identical diagonal and isotropic covariance matrices. So topic component parameters $\boldsymbol{\beta}$ are the K Gaussian distribution means, μ_k, and covariance matrices, $\Sigma_k = \sigma^2\mathbf{I}$. The step 10 in Algorithm 3 are accordingly expanded as followings.

Step 10.1 Sample μ_k: In the current implementation, the proposal distribution for a topic distribution mean is a Gaussian distribution whose parameters are set to be the mean of the corpus

$$\mu_{\mathbf{D}} = \frac{1}{\sum_{d=1}^{D} N^d} \sum_{d=1}^{D} \sum_{n=1}^{N^d} \mathbf{x}_n^d, \tag{2.20}$$

and a wide covariance in the shape of the corpus covariance

$$\Sigma_{\mathbf{D}} = \frac{f}{\sum_{d=1}^{D} N^d} \sum_{d=1}^{D} \sum_{n=1}^{N^d} \left(\mathbf{x}_n^d - \mu_{\mathbf{D}}\right) \left(\mathbf{x}_n^d - \mu_{\mathbf{D}}\right)^T, \tag{2.21}$$

where f is a user set parameter. Thus,

$$\mu_k^\dagger \sim \mathcal{N}(\cdot|\mu_{\mathbf{D}}, \Sigma_{\mathbf{D}}). \tag{2.22}$$

During sampling, the proposed new candidate $\mu_k^\dagger$ will be accepted with the probability

$$a_k = \min\left\{1, \frac{p\left(\mathbf{\Pi}^{(t)}, \mathbf{S}^{(t)}, \mathbf{M}^{(t)}, \mathbf{D}|\mu_k^\dagger\right) \mathcal{N}(\mu_k^{(t-1)}|\mu_{\mathbf{D}}, \Sigma_{\mathbf{D}})}{p\left(\mathbf{\Pi}^{(t)}, \mathbf{S}^{(t)}, \mathbf{M}^{(t)}, \mathbf{D}|\mu_k^{(t-1)}\right) \mathcal{N}(\mu_k^\dagger|\mu_{\mathbf{D}}, \Sigma_{\mathbf{D}})}\right\}. \tag{2.23}$$

Step 10.2 Sampling σ^2: In the current implementation, the topic component distributions are represented with K Gaussian distributions with different means but identical diagonal and isotropic covariance matrices. So for the topic covariance matrix, only the diagonal element σ^2 is needed to sample. The proposal distribution for σ^2 is a uniform distribution, Unif$(0, u)$, with

$$u = f_u \cdot \frac{1}{2}\left\{\max_{\mathbf{x}_n} d^2(\mathbf{x}_n, \mu_{\mathbf{D}}) - \min_{\mathbf{x}_n} d^2(\mathbf{x}_n, \mu_{\mathbf{D}})\right\}, \tag{2.24}$$

where f_u is a user set parameter, which is set to be 1 in the experiments. The corresponding candidates for all the cluster covariance matrices $\mathbf{\Sigma}$ will be

$$\mathbf{\Sigma}^\dagger = \{\Sigma_1^\dagger, \Sigma_2^\dagger, ..., \Sigma_K^\dagger\}, \tag{2.25}$$

where each $\Sigma_k^\dagger = \sigma^{2^\dagger}\mathbf{I}$. In the tth iteration, the proposed new candidate $\mathbf{\Sigma}^\dagger$ will be accepted with the probability

$$a_{\mathbf{\Sigma}} = \min\left\{1, \frac{p\left(\mathbf{\Pi}^{(t)}, \mathbf{S}^{(t)}, \mathbf{M}^{(t)}, \mathbf{D}|\mathbf{\Sigma}^\dagger\right)}{p\left(\mathbf{\Pi}^{(t)}, \mathbf{S}^{(t)}, \mathbf{M}^{(t)}, \mathbf{D}|\mathbf{\Sigma}^{(t-1)}\right)}\right\}. \tag{2.26}$$

2.3.2 Topic as Multinomial Distribution

For topics modeled as multinomial distributions, the proposal distribution for β_k is a Dirichlet distribution, Dir($\mathbf{1}_V$). In the tth iteration, the proposed new candidate $\beta_k^\dagger$ will be accepted with the probability

$$a_\beta = \min\left\{1, \frac{p\left(\mathbf{\Pi}^{(t)}, \mathbf{S}^{(t)}, \mathbf{M}^{(t)}, \mathbf{D}|\beta_k^\dagger\right)}{p\left(\mathbf{\Pi}^{(t)}, \mathbf{S}^{(t)}, \mathbf{M}^{(t)}, \mathbf{D}|\beta_k^{(t-1)}\right)}\right\}. \tag{2.27}$$

Chapter 3

Experiments on Synthetic Data

In this section, a series of experiments on synthetic data are conducted to fully explore the properties of PM-LDA. The experiments begin with estimating memberships with known topics and estimating topic centers with known memberships, and then extend to estimate more unknown parameters including the topic proportion, scaling factor, membership and topics using a Gibbs sampler.

3.1 Membership Estimation with Known Topics

3.1.1 Experiment 1 - Varying the Number of Samples

Suppose that the topic proportion $\boldsymbol{\pi}$ and the scaling factor s are fixed, the PM-LDA is degraded to a mixture model with partial memberships. As discussed in Section 1.3 of Chapter 1, a data point $\mathbf{x}_n$ is sampled from a distribution which is a convex combination of all of the topics, weighted by its membership values $\mathbf{z}_n$. A geometric interpretation is

that the distributions that can generate data lie on the $(K-1)$-simplex with the K topic distributions as its K vertices. An illustration for $K = 3$ is shown in Figure 3.1. The three vertices $\boldsymbol{C}$ represent the three topics in the natural parameter space. For standard mixture model with crisp membership value, only the three vertices can generate data, while for mixture model with partial membership, all the distributions (infinitely many) on the simplex can generate data. In the inference procedure where the topics are known (i.e., C_i are known in Figure 3.1), estimating the membership values is roughly equivalent to estimating the data generating distribution (i.e., dots in Figure 3.1) since the data generating distribution is fully determined by the membership values. Consider a challenging case when N data points are generated by N different data generating distributions. In other words, only one data point is available to estimate the data generating distribution. It can be seen that the estimation accuracy of the membership values is highly dependent on the number of data points.

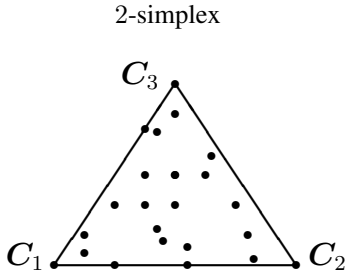


Figure 3.1: The simplex for three topics. C_i represents the natural parameters of the ith topic. The black dots show some of the possible data generating distributions.

This experiment was designed to show the impact of the number of data points on the membership estimation error. It is assumed that all the data points have the same membership values, generated by the same distribution. In the experiment, the membership values are fixed to be $[0.4, 0.6]$. The two Gaussian distributions with $\boldsymbol{\mu}_1 = [2, 2]$, $\boldsymbol{\Sigma}_1 = [0.3, 0.02; 0.02, 0.2]$, and $\boldsymbol{\mu}_2 = [-4, -2]$, $\boldsymbol{\Sigma}_2 = [0.2, 0; 0, 0.4]$ are used as topics. The

data generating distribution is a Gaussian distribution with mean $\boldsymbol{\mu}' = [-2.2030, 0.1317]$, and covariance $\boldsymbol{\Sigma}' = [0.2306, 0.0088; 0.0088, 0.2850]$ according to the following equations,

$$\begin{aligned} \boldsymbol{\Sigma}' &= \left(0.4\boldsymbol{\Sigma}_1^{-1} + 0.6\boldsymbol{\Sigma}_2^{-1}\right)^{-1} \\ \boldsymbol{\mu}' &= \left(0.4\boldsymbol{\mu}_1\boldsymbol{\Sigma}_1^{-1} + 0.6\boldsymbol{\mu}_2\boldsymbol{\Sigma}_2^{-1}\right)\boldsymbol{\Sigma}'. \end{aligned} \tag{3.1}$$

The number of data points is varied from 1 to 582 with stepsize 20. The average squared error of the estimated membership values over 10 runs is shown as the red curve ($F = 1$) in Figure 3.2. The estimated membership values have large error when the number of data point is small.

3.1.2 Experiment 2 - Varying the Covariance Matrices of Topics

With limited number of samples, an important factor on the estimation results is the topic attributes. Assume that the topics are Gaussian distributions. Given the same amount of samples, the larger the Gaussian variance, the bigger the estimation error. In this experiment, the covariance matrices of the two Gaussian topics in Section 3.1.1 are multiplied by a factor F of 1, 4, 9, and 16, respectively. The average squared errors under different multipliers are shown in Figure 3.2. It can be seen that the average squared error increases as the multiplier F becomes larger (the covariance matrix becomes more diffuse).

3.1.3 Experiment 3 - Membership Estimation with Known Topics

In this experiment, the memberships $\mathbf{Z}$ are estimated assuming that the topic proportion $\boldsymbol{\pi}$, the scaling factor s, and the topics $\boldsymbol{\beta}$ are known. The influences of scaling factor and

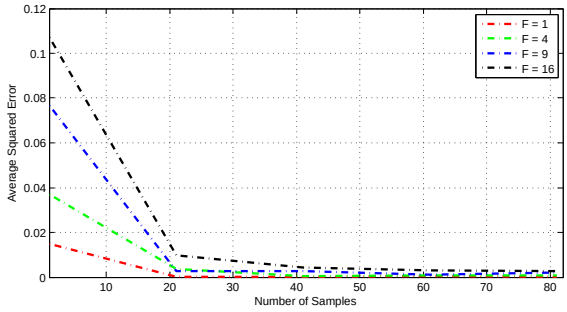


Figure 3.2: Average squared error for different multiplier F ($\Sigma_F = \Sigma \cdot F$). When $N = 1$, the average squared errors for $F = 1, 4, 9, 16$ are 0.015, 0.036, 0.077, and 0.107, respectively.

topic attributes on the membership estimation are investigated. Three types of documents are designed, with sparse (actually binary) memberships, highly mixed memberships and memberships evenly distributed in $[0, 1]$, respectively.

Scaling factor s In this part, the effect of scaling factor s on the membership estimation is investigated using three types of documents, with sparse (actually binary) memberships, highly mixed memberships and memberships evenly distributed in $[0, 1]$, respectively. Each document is generated using two Gaussian topics, with $\boldsymbol{\mu}_1 = [0, 4]$, $\boldsymbol{\Sigma}_1 = 0.01 \times [1, 0; 0, 1]$, and $\boldsymbol{\mu}_2 = [0, -4]$, $\boldsymbol{\Sigma}_2 = 0.01 \times [1, 0; 0, 1]$. Each document has 1000 data points. The topic proportion is fixed to be $[0.5, 0.5]$. The scaling factor s is varied to be 0.001, 0.01, 100, 1000, 5000 and 10000. The estimated memberships are shown in Figure 3.3 - 3.5. As shown in Figure 3.3 and 3.4, the scaling factor s does not effect the estimated membership too much until it increases to 5000. For highly mixed data in Figure 3.5, the scaling factor does not effect the membership estimation result.

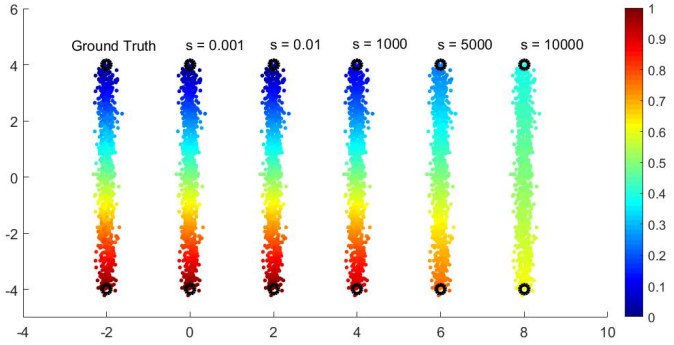


Figure 3.3: Estimated memberships with varying scaling factor on document with membership evenly distributed in range $[0, 1]$. The average squared error of membership estimation when $s = 0.001$, $s = 0.01$, $s = 1000$, $s = 5000$, and $s = 10000$ are 4.0×10^{-4}, 3.9×10^{-4}, 0.0299, 0.0922, and 0.1179, respectively.

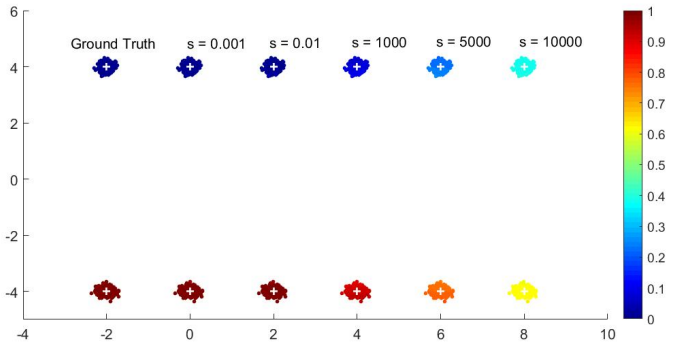


Figure 3.4: Estimated memberships with varying scaling factor on document with crisp memberships. The average squared error of membership estimation when $s = 0.001$, $s = 0.01$, $s = 1000$, $s = 5000$, and $s = 10000$ are 1.7×10^{-5}, 1.8×10^{-5}, 0.1086, 0.2945, and 0.3734, respectively.

Topic Attributes In this part, the effect of topic attributes (i.e., covariance matrix, the distance among topics) on the membership estimation is investigated using three types of documents, with sparse (actually binary) memberships, highly mixed memberships and memberships evenly distributed in $[0, 1]$, respectively. First, the influence of topic covariance matrix is studied. Each document is generated using two Gaussian topics, with $\boldsymbol{\mu}_1 =$

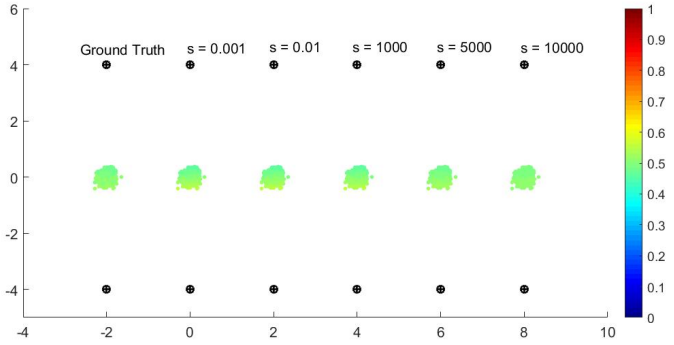


Figure 3.5: Estimated memberships with varying scaling factor on document with highly mixing memberships. The average squared error of membership estimation when $s = 0.001$, $s = 0.01$, $s = 1000$, $s = 5000$, and $s = 10000$ are 3.3×10^{-4}, 3.3×10^{-4}, 1.6×10^{-4}, 1.5×10^{-4}, and 1.7×10^{-4}, respectively.

$[0, 4]$, and $\boldsymbol{\mu}_2 = [0, -4]$. The covariance matrices $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2$ is varied to be $0.01 \times [1, 0; 0, 1]$ and $0.04 \times [1, 0; 0, 1]$. Each document has 1000 data points. The topic proportion is fixed to be $[0.5, 0.5]$. The scaling factor s is varied to be 0.001, 0.01, 100, 1000, 5000 and 10000. The estimated memberships are shown in Figure 3.6 - 3.8. Comparing Figure 3.3 with Figure 3.6, Figure 3.4 and 3.7, and Figure 3.5 and 3.8, it can be seen that as the covariance matrix increases, the membership estimation gets harder.

As shown in Figure 3.4 and 3.7, as the covariance matrix increases, data points with crisp memberships can spread further away from the topic centers. Crisp data points that are located in between the topic centers are often misestimated as data points with mixing partial memberships, thus increasing the membership estimation error. At the same time, as shown in Figure 3.5 and 3.8, data points with partial memberships can also be misestimated as crisp data points. The increase of topic covariance matrix essentially increases the with-in topic scatter and the overlap among topics, thus making topics themselves less distinguishable and decreasing the membership estimation accuracy.

To solve the problem mentioned above, a straightforward solution is to increase the

distance between topics. In the following experiment, the two topic means are varied to be $\boldsymbol{\mu}_1 = 2 \times [0, 4]$, and $\boldsymbol{\mu}_2 = 2 \times [0, -4]$. The estimated memberships are shown in Figure 3.9 - 3.11. Comparing Figure 3.6 with 3.9, Figure 3.7 with 3.10 and Figure 3.8 with 3.11, it can be seen that by increasing the distance between topics, the membership estimation gets more accurate.

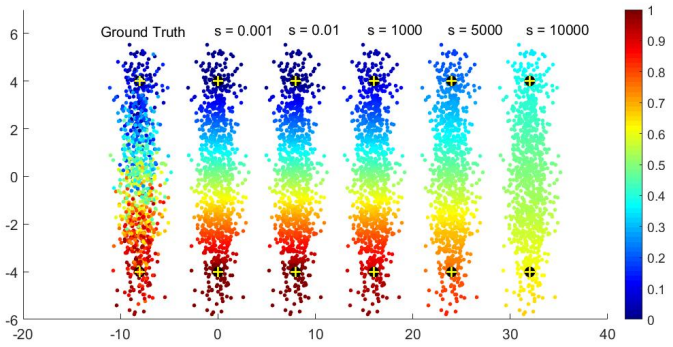


Figure 3.6: Estimated memberships with varying scaling factor on document with membership evenly distributed in range $[0, 1]$. The average squared error of membership estimation when $s = 0.001$, $s = 0.01$, $s = 1000$, $s = 5000$, and $s = 10000$ are 0.0289, 0.0289, 0.0417, 0.0993, and 0.1255, respectively.

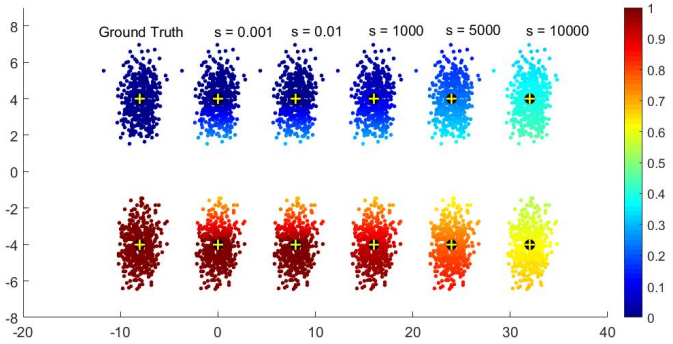


Figure 3.7: Estimated memberships with varying scaling factor on document with crisp memberships. The average squared error of membership estimation when $s = 0.001$, $s = 0.01$, $s = 1000$, $s = 5000$, and $s = 10000$ are 0.0150, 0.0150, 0.1177, 0.2966, and 0.3737, respectively.

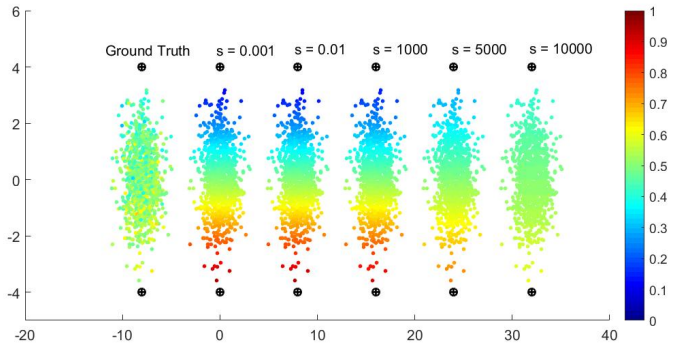


Figure 3.8: Estimated memberships with varying scaling factor on document with highly mixing memberships. The average squared error of membership estimation when $s = 0.001$, $s = 0.01$, $s = 1000$, $s = 5000$, and $s = 10000$ are 0.0305, 0.0305, 0.0114, 0.0049, and 0.0048, respectively.

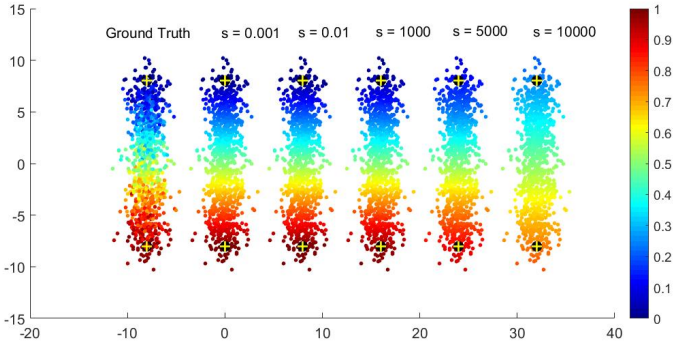


Figure 3.9: Estimated memberships with varying mean on document with membership evenly distributed in range $[0, 1]$. The average squared error of membership estimation when $s = 0.001$, $s = 0.01$, $s = 1000$, $s = 5000$, and $s = 10000$ are 0.0499, 0.0498, 0.0089, 0.0549, and 0.0924, respectively.

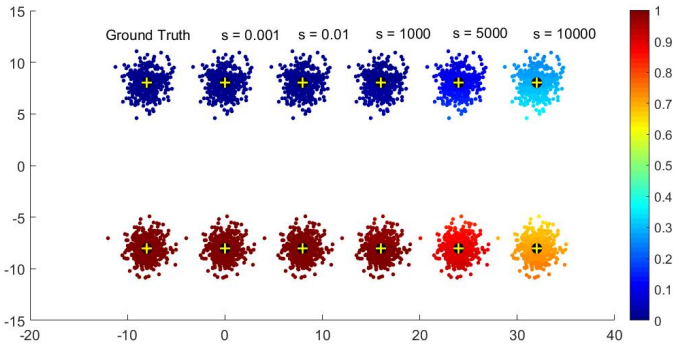


Figure 3.10: Estimated memberships with varying mean on document with crisp memberships. The average squared error of membership estimation when $s = 0.001$, $s = 0.01$, $s = 1000$, $s = 5000$, and $s = 10000$ are 3.8×10^{-6}, 3.8×10^{-6}, 0.0264, 0.1708, and 0.2745, respectively.

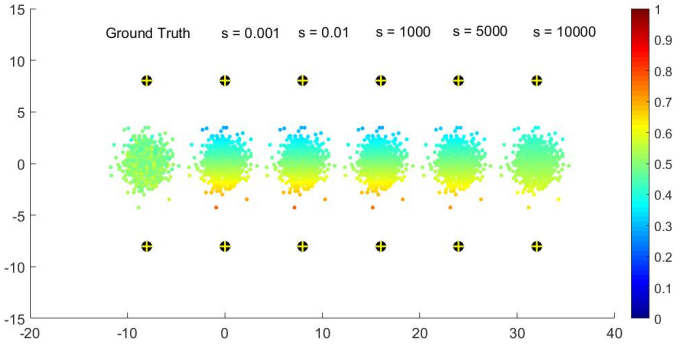


Figure 3.11: Estimated memberships with varying mean on document with highly mixing memberships. The average squared error of membership estimation when $s = 0.001$, $s = 0.01$, $s = 1000$, $s = 5000$, and $s = 10000$ are 0.0320, 0.0320, 0.0109, 0.0022, and 0.0014, respectively.

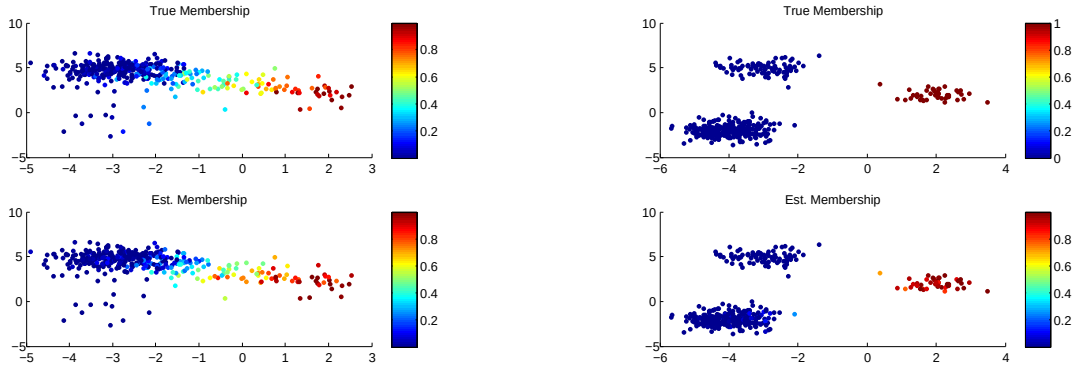
3.1.4 Experiment 4 - Topic Proportion, Scaling Factor, and Membership Estimation with Known Topics

This experiment is a complete inference procedure, which estimates the topic proportion, the scaling factor and the memberships for each document with known topics. The impact of the scaling factor s and the size of topic covariance matrices on the estimation results are investigated.

To test the performance of the Gibbs sampler, an experiment is run on two documents, with sparse (actually binary) and mixing memberships, respectively. The document is generated using three Gaussian topics, with $\boldsymbol{\mu}_1 = [2, 2]$, $\boldsymbol{\Sigma}_1 = 0.4 \times [1, 0; 0, 1]$, $\boldsymbol{\mu}_2 = [-4, -2]$, $\boldsymbol{\Sigma}_2 = 0.4 \times [1, 0; 0, 1]$, and $\boldsymbol{\mu}_3 = [-3, 5]$, $\boldsymbol{\Sigma}_3 = 0.4 \times [1, 0; 0, 1]$. Each document has 400 data points. The estimated topic proportions, scaling factors, and memberships in topic 1 are shown in Figure 3.12. Note that the generation of the binary document doesn't exactly follow the proposed generative process in which it's very unlikely to get a document with all the words having binary memberships. In the generation of the binary document, the original continuous membership vector is converted to a binary one by reassigning the largest element in the vector to be 1 and others to be zeros. The estimation results of the topic proportion and the scaling factor are shown in Table 3.12c. The experimental results show that PM-LDA performs well on both crisp and mixing data points.

Moreover, to investigate the impact of the scaling factor s and the size of topic covariance matrices on the estimation results, the experiments are extended by including the following two sets of experiments.

Varying scaling factor: As discussed in Section 3.1.1, estimating the membership values in the inference procedure is roughly equivalent to estimating the data generating



(a) Mixing, 3 topics

(b) Binary, 3 topics

Doc.	Mixing	Binary
True $\boldsymbol{\pi}$	$[0.1853, 0.0409, 0.7738]$	$[0.1367, 0.6227, 0.2406]$
Est. $\boldsymbol{\pi}$	$[0.1662, 0.0907, 0.7431]$	$[0.1249, 0.6417, 0.2334]$
True s	1.0	N/A
Est. s	1.1	N/A

(c) Estimated topic proportions and scaling factors

Figure 3.12: Estimated topic proportions, scaling factors, and memberships with known topics. (a) and (b) show the true and estimated memberships in the 1st topic. For (a), the ASE of the estimated memberships is 2.9×10^{-2}. For (b), the ASE of the estimated memberships is 1.1×10^{-2}. (c) lists the true and estimated topic proportions and scaling factors.

distributions. For $\mathbf{z} \sim \text{Dir}(s\boldsymbol{\pi})$, when the concentration parameter $s\boldsymbol{\pi}_k$ below 1, the smaller the scaling factor s, the less diverse the membership vectors, and the fewer the data generating distributions. Given a fixed total number of words, there will be more words used for the estimation of each data generating distribution, thus the membership estimation will be more accurate. While when the concentration parameter $s\boldsymbol{\pi}_k$ is above 1, the situation is opposite. The larger the scaling factor s, the more concentrated the membership vectors around $\boldsymbol{\pi}$, thus the membership estimation will be more accurate.

In this experiment, five documents are generated using three Gaussian topics, with $\boldsymbol{\mu}_1 = [2, 2]$, $\boldsymbol{\Sigma}_1 = F \times [1, 0; 0, 1]$, $\boldsymbol{\mu}_2 = [-4, -2]$, $\boldsymbol{\Sigma}_2 = F \times [1, 0; 0, 1]$, and $\boldsymbol{\mu}_3 = [-3, 5]$,

Table 3.1: Average squared error (standard deviation) : varying the scaling factor s

s	$\mathbf{Z}$	$\boldsymbol{\pi}$	s
0.2	$1.6 \times 10^{-2}(5.2 \times 10^{-3})$	$3.8 \times 10^{-3}(2.1 \times 10^{-3})$	$3.5 \times 10^{-1}(6.2 \times 10^{-2})$
1.0	$3.8 \times 10^{-2}(7.0 \times 10^{-3})$	$1.5 \times 10^{-3}(1.2 \times 10^{-3})$	$3.7 \times 10^{-3}(2.3 \times 10^{-3})$
10	$8.2 \times 10^{-2}(1.1 \times 10^{-2})$	$7.0 \times 10^{-4}(6.2 \times 10^{-4})$	$2.8(1.2)$
100	$7.1 \times 10^{-2}(8.5 \times 10^{-3})$	$2.8 \times 10^{-4}(4.2 \times 10^{-4})$	$6.7 \times 10^{3}(2.5 \times 10^{2})$
1000	$6.5 \times 10^{-2}(9.6 \times 10^{-3})$	$1.2 \times 10^{-4}(6.4 \times 10^{-5})$	$9.6 \times 10^{5}(5.3 \times 10^{3})$

$\boldsymbol{\Sigma}_3 = F \times [1, 0; 0, 1]$, where $F = 0.4$. Each of the five documents has 400 words, and their scaling factors are 0.2, 1, 10, 100, and 1000 respectively. The hyper-parameter $\boldsymbol{\alpha}$ is set to be $[1, 1, 1]$, and λ is set to be 0.5. The average squared error and standard deviation of the estimated variables over 5 runs are shown in Table 3.1. As shown in the first column in Table 3.1, the average squared error of the estimated membership first increases as the scaling factor increases from 0.2 to 10, and then decreases as the scaling factor further increases to 100 and 1000. When $s = 0.2$ and $s = 1$, $s\boldsymbol{\pi}$ is below 1. The smaller the scaling factor s, the more crisp the memberships, the more accurate the membership estimation. When $s = 10$, $s = 100$ and $s = 1000$, $s\boldsymbol{\pi}_k$ is above 1. The membership estimation error becomes smaller as the scaling factor s increases because the memberships become more concentrated on $\boldsymbol{\pi}$, more similar to each other.

As for the estimation of the topic proportion $\boldsymbol{\pi}$, its estimation accuracy increases as the scaling factor s increases, since the membership vectors becomes more and more concentrated on $\boldsymbol{\pi}$. For the estimation of the scaling factors, the significant increase of the estimation error when $s = 100$ and $s = 1000$ is due to its proposal distribution, $p(s) = \lambda e^{-\lambda s}(\lambda = 0.5)$, where the largest candidate proposed in this experiment is around 20, and no candidates approach 100 or 1000.

Table 3.2: Average squared error (standard deviation) : varying the topic covariance matrices

F	0.016	0.4	4.0
$\mathbf{Z}$	$1.6 \times 10^{-3}(2.9 \times 10^{-4})$	$3.8 \times 10^{-2}(7.0 \times 10^{-3})$	$3.2 \times 10^{-1}(6.1 \times 10^{-2})$
$\boldsymbol{\pi}$	$1.9 \times 10^{-3}(9.6 \times 10^{-4})$	$1.5 \times 10^{-3}(1.2 \times 10^{-3})$	$1.8 \times 10^{-3}(2.0 \times 10^{-3})$
s	$4.7 \times 10^{-2}(2.8 \times 10^{-2})$	$3.7 \times 10^{-3}(2.3 \times 10^{-3})$	$2.9 \times 10^{-2}(1.4 \times 10^{-2})$

Varying topic covariance matrices: As discussed in Section 3.1.2 and 3.1.3, another important factor on the membership estimation is the topic itself. The larger the topic covariance matrices, the lower the membership estimation accuracy. In this experiment, three documents are used. The first document is the one used in the above experiment with $s = 1$ and $F = 0.4$. The second document is generated using topics with the same $\boldsymbol{\mu}_1$, $\boldsymbol{\mu}_2$, and $\boldsymbol{\mu}_3$ as in the above experiment and covariance matrices with $F = 0.016$. The third document is generated using topics with the same $\boldsymbol{\mu}_1$, $\boldsymbol{\mu}_2$, and $\boldsymbol{\mu}_3$ as in the above experiment and covariance matrices with $F = 4.0$. The average squared error and standard deviation of the estimated variables over 5 runs are shown in Table 3.2. As shown in the first row in Table 3.2, the average squared error of the estimated membership increases as the covariance matrices become larger, which verifies the discussion in Section 3.1.2 and 3.1.3.

3.2 Parameter Estimation for Topics with Diagonal and Isotropic Covariance Matrices

In this section, the memberships $\mathbf{Z}^d$, the topic proportions $\boldsymbol{\pi}^d$, the scaling factor s^d, and the topics are assumed to be unknown, and will be estimated in the experiments. Five documents are generated using two Gaussian topics, with $\boldsymbol{\mu}_1 = [2, 2]$, $\boldsymbol{\Sigma}_1 = 0.4 \times [1, 0; 0, 1]$,

and $\boldsymbol{\mu}_2 = [-4, -2]$, $\boldsymbol{\Sigma}_2 = 0.4 \times [1, 0; 0, 1]$. The hyper-parameters $\boldsymbol{\alpha}$ and λ is set to be $\mathbf{1}_K$ and 0.5, respectively. Three corpora are reconstructed, containing the first 3, 4, and 5 documents in the generated 5 documents, respectively. The experiments are run for five times, and the experimental results are shown Table 3.3 - 3.5. One example of the estimation results is shown in Figure 3.13 and Table 3.6. As shown in Table 3.3, the topic estimation accuracy increases as the total number of words increases from 1200 to 2000. For the topic proportion and membership estimation, as shown in Table 3.4 and 3.5, the increase in the number of words does not make much difference to the topic proportion or the membership estimation.

Moreover, in this experiment, the proposed PM-LDA is compared to the standard LDA model and FCM. For FCM, the fuzzifier is varied from 1.05 to 3.00 with stepsize 0.05. The FCM results with the smallest topic estimation error is selected for comparison. The best fuzzifier for the three Corpora are 2.30, 2.30, and 2.60, respectively. Their performance comparisons are illustrated in Figure 3.14 and listed in Table 3.3 - 3.5. Note that for FCM, the estimation on topic covariance and topic proportion is not available. For the topic estimation (in Table 3.3), It can be seen that the PM-LDA has the best performance. For the topic proportion estimation (in Table 3.4), the PM-LDA outperforms the standard LDA model on all the documents except document D2. Since FCM doesn't treat each document individually, its membership estimation is evaluated based on the whole corpus. As shown in Table 3.5, the PM-LDA outperforms the standard LDA, while FCM outperforms the PM-LDA slightly. As shown in Figure 3.13, the synthetic data actually also fit the spherical prototypes assumed by FCM well. It is possible that FCM can outperform PM-LDA in membership estimation in such scenario.

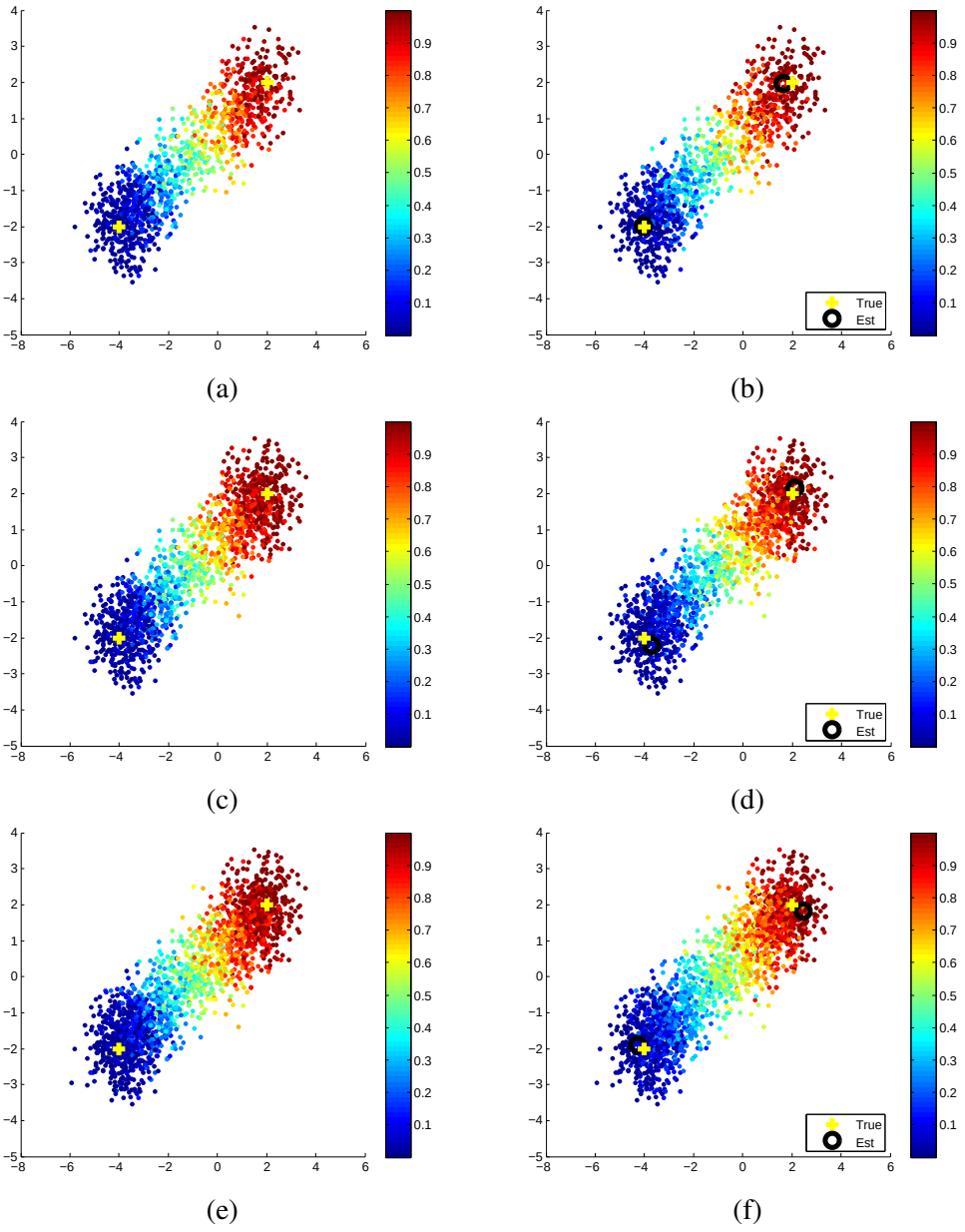
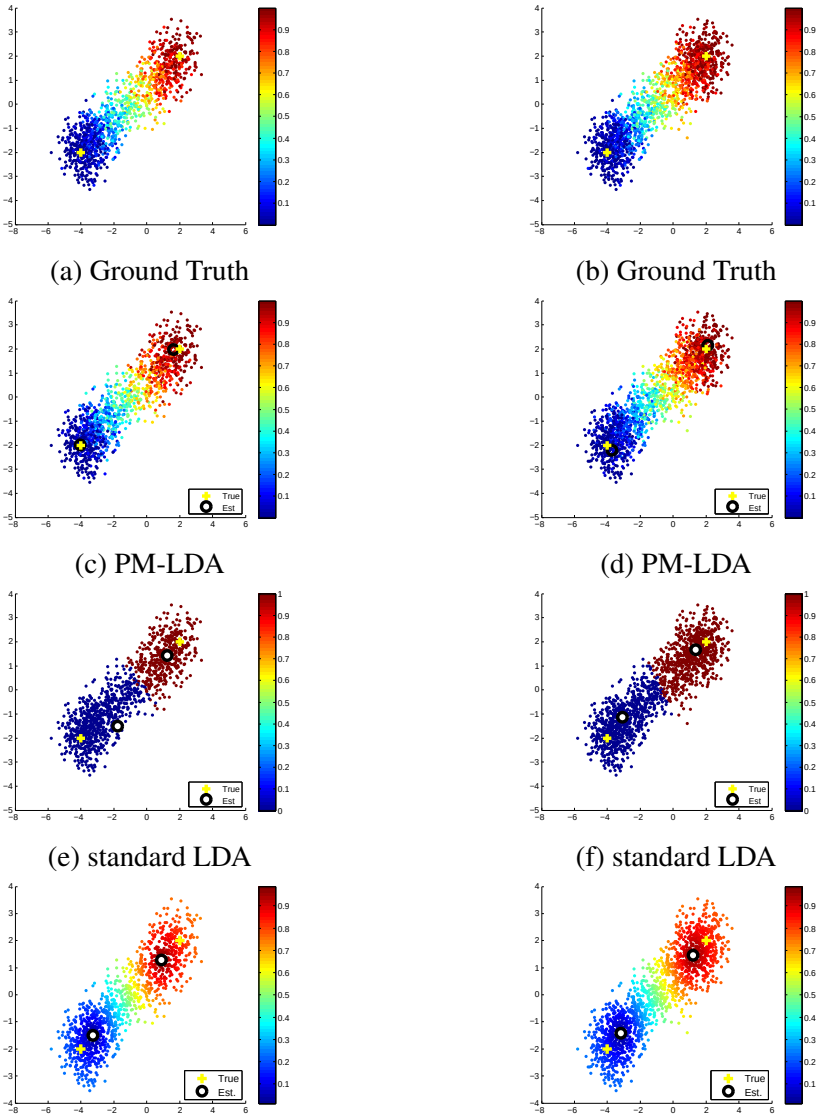


Figure 3.13: An example of the estimated memberships and topics for different corpora. The left column shows the true membership and topic centers, and the right column shows the estimated membership and topic centers. The ASE of memberships are 2.54×10^{-2}, 2.33×10^{-2}, and 2.46×10^{-2}, respectively. The estimated topic covariance matrices are $0.40 \cdot \mathbf{I}$, $0.42 \cdot \mathbf{I}$, and $0.44 \cdot \mathbf{I}$, respectively.



(a) Ground Truth (b) Ground Truth

(c) PM-LDA (d) PM-LDA

(e) standard LDA (f) standard LDA

(g) FCM (fuzzifier = 2.3) (h) FCM (fuzzifier = 2.3)

Figure 3.14: An example of the estimated memberships and topics for different corpora using different methods. The first row shows the true membership and topic centers. Row $2-4$ show the estimated membership and topic centers using the PM-LDA with partial membership, the standard LDA model, and the FCM method, respectively.

Table 3.3: Average squared error (standard deviation) of the estimated topics for the three corpora

Topics		Corpus 1	Corpus 2	Corpus 3
mean	PM-LDA	$6.7 \times 10^{-1}(5.0 \times 10^{-1})$	$1.9 \times 10^{-1}(1.7 \times 10^{-1})$	$1.8 \times 10^{-1}(9.6 \times 10^{-2})$
	LDA	$2.0 \times 10^{0}(5.2 \times 10^{-1})$	$1.2 \times 10^{0}(6.9 \times 10^{-1})$	$1.5 \times 10^{0}(5.8 \times 10^{-1})$
	FCM	$1.2 \times 10^{0}(3.1 \times 10^{-6})$	$1.0 \times 10^{0}(1.9 \times 10^{-6})$	$1.0 \times 10^{0}(1.9 \times 10^{-6})$
cov.	PM-LDA	$5.3 \times 10^{-3}(2.9 \times 10^{-3})$	$4.6 \times 10^{-3}(5.4 \times 10^{-3})$	$3.6 \times 10^{-3}(2.7 \times 10^{-3})$
	LDA	$9.0 \times 10^{-2}(0)$	$9.0 \times 10^{-2}(0)$	$9.0 \times 10^{-2}(0)$
	FCM	N/A	N/A	N/A

Table 3.4: Average squared error (standard deviation) of the estimated topic proportion for the three corpora

Document		Corpus 1	Corpus 2	Corpus 3
D1	PM-LDA	$1.3 \times 10^{-3}(9.0 \times 10^{-4})$	$2.3 \times 10^{-3}(3.5 \times 10^{-3})$	$4.0 \times 10^{-4}(8.0 \times 10^{-4})$
	LDA	$4.9 \times 10^{-3}(5.6 \times 10^{-3})$	$1.4 \times 10^{-3}(1.9 \times 10^{-3})$	$2.4 \times 10^{-3}(1.8 \times 10^{-3})$
D2	PM-LDA	$1.6 \times 10^{-3}(1.2 \times 10^{-3})$	$3.9 \times 10^{-3}(4.5 \times 10^{-3})$	$7.0 \times 10^{-4}(1.2 \times 10^{-3})$
	LDA	$4.9 \times 10^{-3}(4.1 \times 10^{-3})$	$2.0 \times 10^{-3}(2.3 \times 10^{-3})$	$1.3 \times 10^{-3}(2.1 \times 10^{-3})$
D3	PM-LDA	$2.2 \times 10^{-3}(2.7 \times 10^{-3})$	$1.3 \times 10^{-3}(1.6 \times 10^{-3})$	$2.5 \times 10^{-3}(2.7 \times 10^{-3})$
	LDA	$1.2 \times 10^{-2}(1.3 \times 10^{-2})$	$3.9 \times 10^{-3}(3.7 \times 10^{-3})$	$7.8 \times 10^{-3}(1.5 \times 10^{-2})$
D4	PM-LDA	N/A	$2.0 \times 10^{-4}(2.0 \times 10^{-4})$	$1.5 \times 10^{-3}(2.6 \times 10^{-3})$
	LDA	N/A	$3.9 \times 10^{-3}(1.6 \times 10^{-3})$	$8.5 \times 10^{-3}(6.2 \times 10^{-3})$
D5	PM-LDA	N/A	N/A	$5.0 \times 10^{-4}(4.0 \times 10^{-4})$
	LDA	N/A	N/A	$2.8 \times 10^{-3}(4.3 \times 10^{-3})$

Table 3.5: Average squared error (standard deviation) of the estimated membership for the three corpora

Document		Corpus 1	Corpus 2	Corpus 3
D1	PM-LDA	$8.6 \times 10^{-2}(3.2 \times 10^{-2})$	$9.8 \times 10^{-2}(4.0 \times 10^{-2})$	$1.0 \times 10^{-1}(5.5 \times 10^{-2})$
	LDA	$2.8 \times 10^{-1}(1.3 \times 10^{-1})$	$2.8 \times 10^{-1}(2.6 \times 10^{-1})$	$2.7 \times 10^{-1}(1.1 \times 10^{-1})$
D2	PM-LDA	$8.1 \times 10^{-2}(3.0 \times 10^{-2})$	$8.9 \times 10^{-2}(2.8 \times 10^{-2})$	$9.4 \times 10^{-2}(4.5 \times 10^{-2})$
	LDA	$2.6 \times 10^{-1}(1.2 \times 10^{-1})$	$2.3 \times 10^{-1}(2.0 \times 10^{-1})$	$2.9 \times 10^{-1}(1.0 \times 10^{-1})$
D3	PM-LDA	$7.9 \times 10^{-2}(2.7 \times 10^{-2})$	$9.2 \times 10^{-2}(3.1 \times 10^{-2})$	$9.2 \times 10^{-2}(4.7 \times 10^{-2})$
	LDA	$2.8 \times 10^{-1}(1.5 \times 10^{-1})$	$2.3 \times 10^{-1}(1.7 \times 10^{-1})$	$2.8 \times 10^{-1}(1.1 \times 10^{-1})$
D4	PM-LDA	N/A	$7.7 \times 10^{-2}(3.3 \times 10^{-2})$	$6.9 \times 10^{-2}(3.6 \times 10^{-2})$
	LDA	N/A	$1.5 \times 10^{-1}(7.5 \times 10^{-2})$	$3.2 \times 10^{-1}(1.8 \times 10^{-1})$
D5	PM-LDA	N/A	N/A	$1.0 \times 10^{-1}(5.4 \times 10^{-2})$
	LDA	N/A	N/A	$2.9 \times 10^{-1}(1.0 \times 10^{-1})$
FCM		$2.9 \times 10^{-2}(5.6 \times 10^{-8})$	$2.6 \times 10^{-2}(3.1 \times 10^{-8})$	$2.4 \times 10^{-2}(2.5 \times 10^{-8})$

		Corpus 1	**Corpus 2**	**Corpus 3**
Document	True Mem.	Est. Mem.	Est. Mem.	Est. Mem.
D1	$[0.3945, 0.6055]$	$[0.4092, 0.5908]$	$[0.3811, 0.6189]$	$[0.3871, 0.6129]$
D2	$[0.4018, 0.5982]$	$[0.4122, 0.5878]$	$[0.4092, 0.5908]$	$[0.3945, 0.6055]$
D3	$[0.4771, 0.5229]$	$[0.4918, 0.5082]$	$[0.4420, 0.5580]$	$[0.4631, 0.5369]$
D4	$[0.7365, 0.2635]$	N/A	$[0.6994, 0.3006]$	$[0.6937, 0.3063]$
D5	$[0.3784, 0.6216]$	N/A	N/A	$[0.3652, 0.6348]$

Table 3.6: An example of the estimated topic proportions for the three corpora

3.3 Parameter Estimation for Topics with Full Covariance Matrices

In section 3.2, the topic covariance matrices are assumed to be diagonal and isotropic. In this section, this assumption is dropped and the topic covariance matrices are considered as full covariance matrices. The proposal distribution for covariance matrix is Wishart distribution, defined as

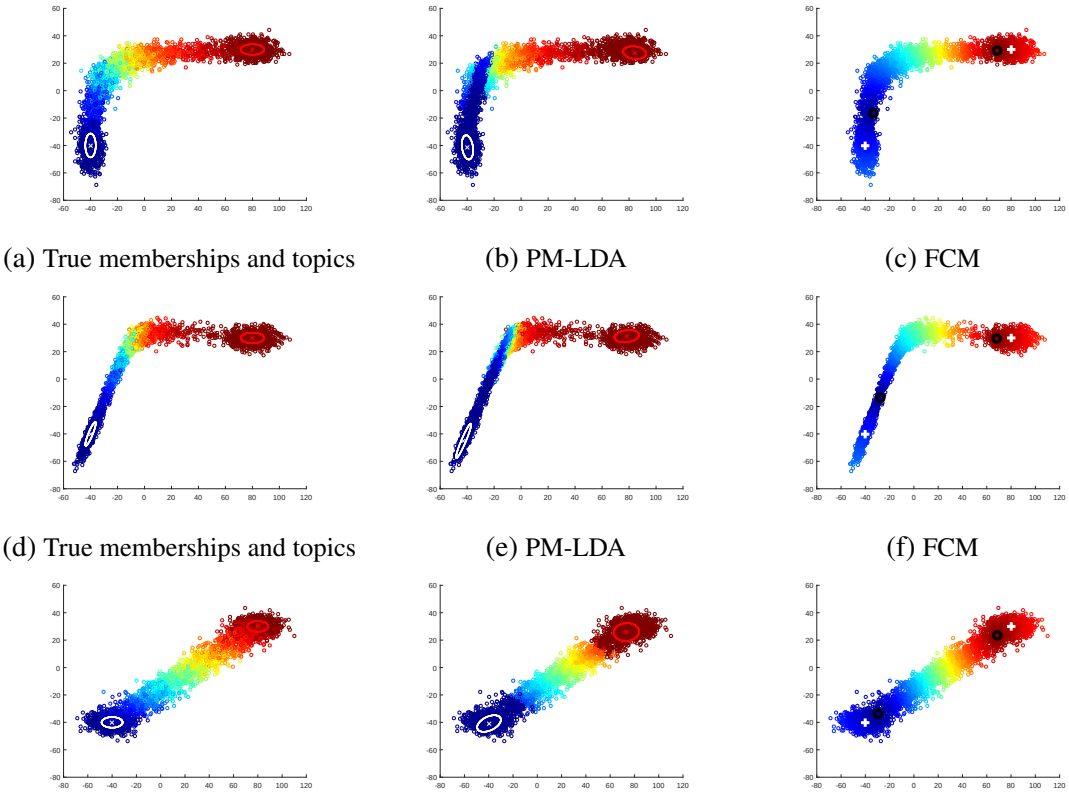
$$p(\mathbf{\Sigma}, \mathbf{\Sigma}_w, v) = \frac{|\mathbf{\Sigma}|^{(v-d-1)/2} \exp\left(-\frac{1}{2}\text{trace}(\mathbf{\Sigma}_w^{-1}\mathbf{\Sigma})\right)}{2^{vd/2}\pi^{d(d-1)/4}|\mathbf{\Sigma}_w|^{v/2}\Gamma(v/2)\cdots\Gamma(v-(d-1))/2}, \tag{3.2}$$

where v is called the degrees of freedom parameter and $\mathbf{\Sigma}_w v$ is the mean of the distribution.

In this experiment, three sets of synthetic data using different full covariance matrices are generated. Each set of synthetic data has 4 documents, each with 500 data points. The means are fixed to be $[80, 30]$ and $[-40, -40]$. The covariance matrices are varied to be $\Sigma_1 = [75, 0; 0, 15]$, $\Sigma_2 = [15, 0; 0, 75]$; $\Sigma_1 = [75, 0; 0, 15]$, $\Sigma_2 = [75, 30; 30, 15]$; and $\Sigma_1 = [75, 0; 0, 15]$, $\Sigma_2 = [75, 0; 0, 15]$;

The three sets of synthetic data are shown in the first column in Figure 3.15. The results show that the angle between two covariance matrices greatly controls the shape (distribution) of synthetic data. For covariance matrix $\mathbf{\Sigma}_k$, the parameter $\mathbf{\Sigma}_w$ in the wishart distribution is set to be the covariance of data points that are labeled as k by k-means, scaled by 0.5. The parameter v controls the sampling variability. The sampling variability is large when v is small. In the experiment, v is set to be 2. FCM is also applied to the three data sets for comparison with PM-LDA. The *fuzzifier* m is varied to be 1.1 to 3.0 with increment 0.1. The experiment is run for ten times. For FCM, in each run, the best membership estimation result is selected for comparison with PM-LDA. One example of

the parameter estimation results are shown in Figure 3.15. The proposed PM-LDA can achieve higher estimation accuracy than FCM for full covariance matrices.



(a) True memberships and topics (b) PM-LDA (c) FCM

(d) True memberships and topics (e) PM-LDA (f) FCM

(g) True memberships and topics (h) PM-LDA (i) FCM

Figure 3.15: Parameter Estimation results of synthetic data with full covariance matrices. For PM-LDA, the average squared membership error ($\pm$standard deviation) for the three datasets over ten runs are $0.02095(\pm 0.0049)$, $0.0365(\pm 0.0050)$, and $0.0107(\pm 0.0013)$, respectively. For FCM, the average squared membership error ($\pm$standard deviation) for the three datasets over ten runs are $0.04980(\pm 0.00)$, $0.0663(\pm 0.00)$, and $0.0132(\pm 0.00)$, respectively.

3.4 Convergence

In this section, I evaluate the convergence rate of the proposed MCMC sampling method. Four sets of data points are generated using two Gaussian clusters, $\mu_1 = [2, 2]$, $\mathbf{\Sigma}_1 = [0.1, 0; 0, 0.1]$, and $\mu_2 = [-4, -2]$, $\mathbf{\Sigma}_2 = [0.1, 0; 0, 0.1]$, with 1000, 2000, 3000, and 4000 data points, respectively. A plot that keeps track of the energy vs. computation time is produced. x-axis denotes the number of iterations and y-axis denotes the current energy relative to the largest energy [47]. The convergence rate for the four datasets are shown in Figure 3.16. As shown in the figure, the energy ratio decreases to 4 within in 500 iterations and arrives at the largest energy within 5000 iterations. As the number of data points increases, the convergence rate becomes slower. The experiment is run for 5 times on another data set with 2500 data points. The convergence plot is shown in Figure 3.17, where the energy ratio drops to 4 within 200 iterations.

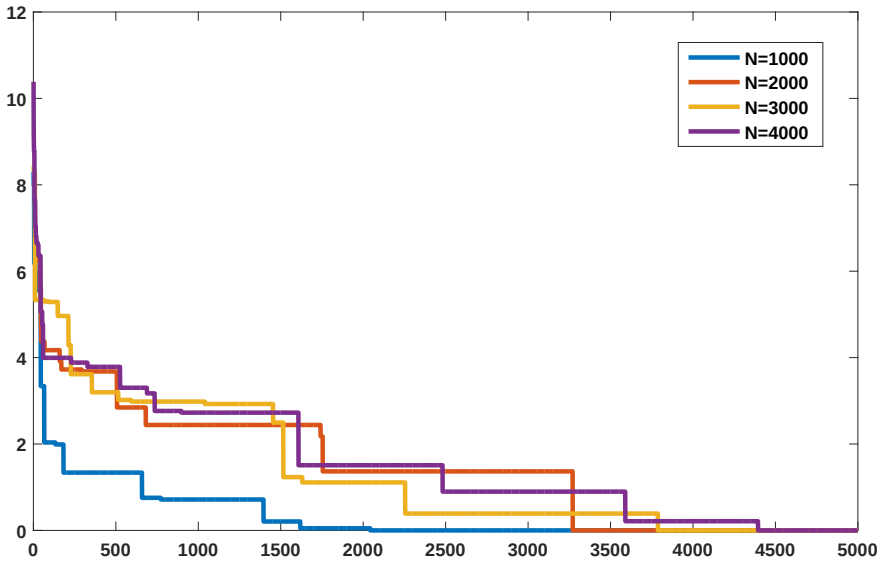


Figure 3.16: Energy vs. run time plot. x-axis denotes the iteration number and y-axis denotes the log of the energy ratio.

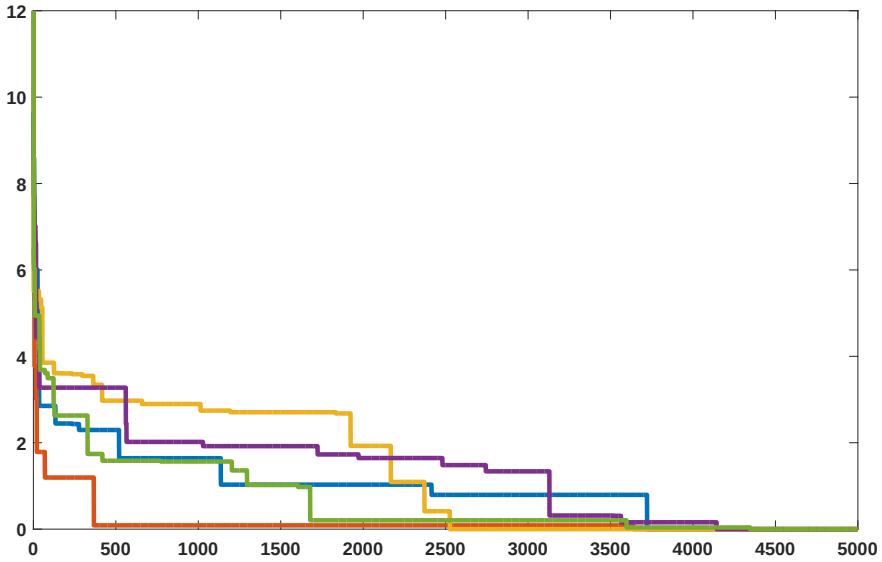


Figure 3.17: Energy vs. run time plot when $N = 2500$.

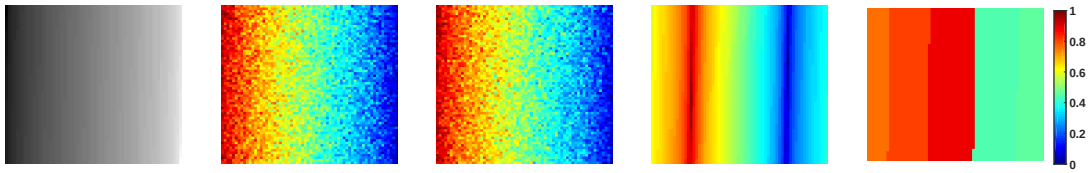
3.5 Experiments on Synthetic Images

Common datasets for crisp segmentation (i.e., MSRC) are unsuitable for evaluation of PM-LDA as they do not include transition regions. Also, the provided segmentation ground truth is only labeled for crisp segmentation. For quantitative comparison with other segmentation methods, ten 64×64 synthetic images (in which the true partial membership values are known) are used in this experiment. The gray-scale pixel values are randomly generated for each image using two Gaussian topics. As the gray-scale pixel value ranges from 0 to 255, the two Gaussian topics are set to be $\mathcal{N}(64, 64)$ and $\mathcal{N}(192, 64)$. For each image, 4096 samples are generated, sorted, and reshaped into a 64×64 matrix. The negative sample is reset to be 0 and the sample greater than 255 is reset to be 255. One of the

generated synthetic images is shown in Figure 3.18a.

For LDA, it is possible to compute the labeling confidence map by normalizing the label posterior density given the observations (Equation (1.11)). However, the posterior is conceptually very different from partial memberships. LDA assumes there is one true class per point (and the posterior provides the probability for each). Instead, the underlying assumption of a partial membership model is that each data point is a *mixture* of topics. Therefore, equal LDA posterior probabilities say that a point is equally likely to have been generated from either topic but not by a mixture of the topics.

The average squared error($\pm$standard deviation) of partial membership estimation using PM-LDA, FCM and LDA over 5 runs are $0.006(\pm 0.001)$, $0.169(\pm 0.163)$ and $281.2(\pm 20.8)$, respectively. An experimental result is shown in Figure 3.18. For quantitative evaluation, only synthetic imagery is used since true partial memberships are unknown and cannot be assigned in a rigorous or feasible way for real imagery.



(a) (b) (c) (d) (e)

Figure 3.18: Partial membership estimation on a synthetic image: (a) the image (b) true partial memberships (c) PM-LDA result (d) FCM result (f) normalized LDA posterior result

Chapter 4

Experiments on Real Imagery

In this section, results of image segmentation on three datasets: (i) a Synthetic Aperture SONAR (SAS) image dataset, (ii) a Fog-Mountain dataset, and the (iii) MSRC dataset [48] are shown. In (i) and (ii), PM-LDA is compared with FCM and LDA to show the ability of PM-LDA to handle partial memberships. In (iii), PM-LDA is compared with LDA to show that PM-LDA also works on 0-1 membership values (i.e., crisp partitioning). For the LDA implementation, the MATLAB topic modeling toolbox 1.4 [5] is used.

4.1 Synthetic Aperture Sonar Imagery

Synthetic aperture sonar systems differ from conventional side-scan sonar where, instead of each ping echo return being processed independently, SAS systems coherently combine the returns from multiple sonar pings along a known track to synthesize a large acoustic aperture. This allows for the ability to produce high resolution images at low frequencies and a large range [49–51]. SAS imagery have resolutions down to the centimeter scale and

range values up to hundreds of meters [51]. In SAS imagery natural textures such as sand ripples and sea grass, and small objects are now identifiable, allowing the sonar imagery to be used for seabed segmentation and classification [52, 53], target detection [54, 55], and other applications. Figure 4.1 shows examples of different seafloor types present in SAS imagery.

As discussed in Section 1.1.4, the assumption in the standard LDA that each word comes from one and only one topic is not necessarily perfect for image documents. For example, in Figure 4.1, the gradual transition from one seabed type to another one results in a fuzzy boundary, where the visual words have no crisp memberships. To tackle this problem, the PM-LDA model is proposed, which is applicable to both SONAR imagery and visual natural imagery.

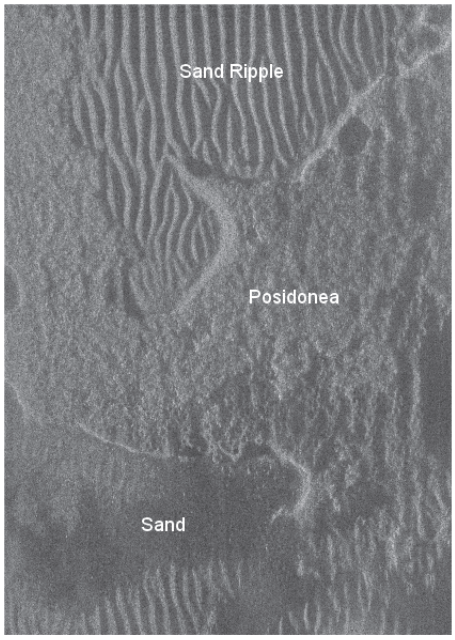


Figure 4.1: A SAS image containing sand ripples, posidonea (a type of sea grass), and hard packed sand.

SAS imagery are generally analyzed by considering features that describe image inten-

sity and texture such as the mean, the variance, etc. However, given a particular sensing geometry and bathymetry, the texture and pixel values imaged by a SAS system can vary drastically over the same spatial region of the sea-floor. Features based on image characteristics are often not qualifying invariant parameters of the sea-floor but, instead, are describing the relationship of that particular sensing geometry with respect to that data of the sea-floor. A different pass of the SAS sensor with varying range and aspects over the same region may result in very different image-based features. In the previous work (discussed in Appendices A and B), several methods have been developed to recovery the underlying bathymetry depicted in a SAS imagery, and characterize the sand ripple with sand ripple frequency, amplitude and orientation, and other seafloor types with Gaussian Markov random fields (GMRFs). Appendix A presents a method for estimating sand ripple frequency, amplitude, and orientation values from a single SAS image as well as from sets of SAS imagery over an area using a hierarchical Bayesian framework and a known sensing geometry. This is accomplished through the development of an extended model for sand ripple characterization and a Metropolis-within-Gibbs sampler to estimate sand ripple frequency, amplitude, and orientation characteristics for multi-aspect high-frequency side-look sonar data. Appendix B investigates the use of GMRFs to characterize the underlying bathymetry depicted in a SAS image. The GMRF parameters that describe the bathymetry of a region are estimated using an Alternating Optimization (AO) with Iterated Conditional Modes (ICM) method where the GMRF parameters and the estimated bathymetry map are updated alternately.

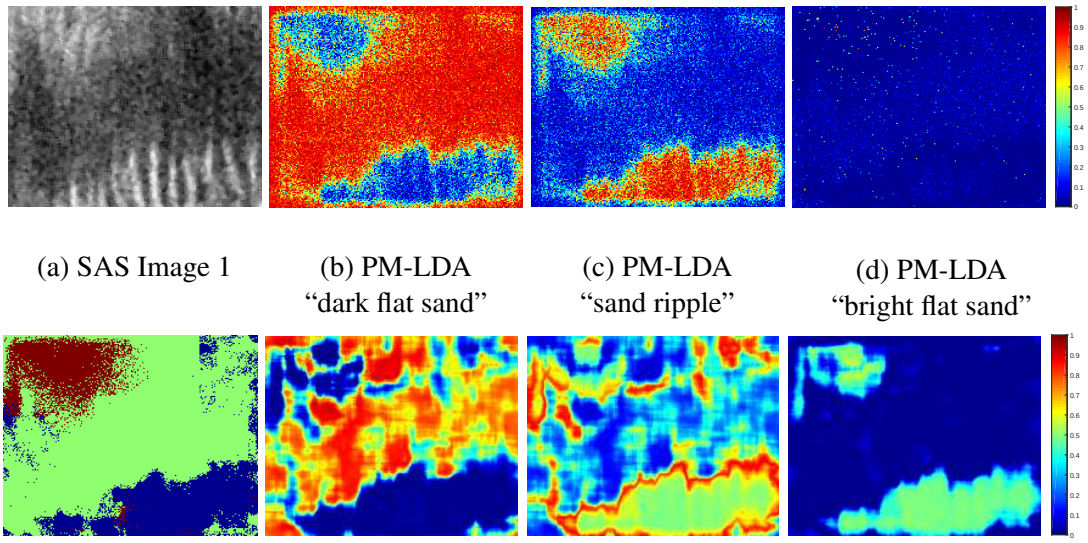
4.2 Experiments on Real SAS imagery

4.2.1 Using Sliding Window as Document

Synthetic Aperture Sonar (SAS) Imagery Dataset From our SAS image dataset, 4 images (shown in subfigure (a) of Figure 4.2 - 4.5) are selected and their average intensity value and entropy within a 21 $\times$ 21 window are computed as feature values. The average intensity value is scaled up ($\times 10$) to the same magnitude of entropy. Each image is divided into multiple documents using a sliding window approach. A document consists of all of the feature vectors associated with each pixel (i.e., visual words) in the window. The number of topics in this dataset is set to 3. For LDA, a dictionary of size 100 is built by clustering all the computed feature values using the K-means. FCM results with $m = 1.5$ are shown as it provided the best results. Due to the lack of ground-truth of this dataset, the qualitative segmentation results are shown in Figure 4.2 - 4.5. Subfigures (b), (c), and (d) show the partial membership map in "dark flat sand" , "sand ripple" , and "bright flat sand" topic using PM-LDA, respectively. Subfigures (f), (g), and (h) show the partial membership map in each of the three clusters using FCM, respectively. In (b) - (d) and (f) - (h), the color indicates the degree of membership of a visual word in a topic. Red corresponds to a full membership of 1 and dark blue color corresponds to a membership value of 0. The LDA result is shown in (e) where color indicates topic.

From the experimental results, it can be seen that PM-LDA achieves much better results than FCM and LDA. As shown in Figure 4.2c and 4.3c, the segmentation results of PM-LDA show a gradual change from "sand ripple" to "dark flat sand". The color is gradually changed from red, orange, cyan to dark blue. FCM captures the gradual transition to some

extent but is not able to clearly differentiate between clusters. For example, as shown in Figure 4.3g - 4.3h and Figure 4.4g - 4.4h, using FCM, the rippled region in Images 2 and 3 is assigned to 2 clusters with nearly equal partial memberships. As LDA cannot generate partial memberships, in Figure 4.2e and 4.3e, Image 1 and 2 are simply partitioned into different topics using LDA. Yet, by comparing Figure 4.4e with 4.4d and Figure 4.5e with 4.5d, it can be seen that on Image 3 and 4 with insignificant transition regions, LDA achieves similar segmentation result to PM-LDA.

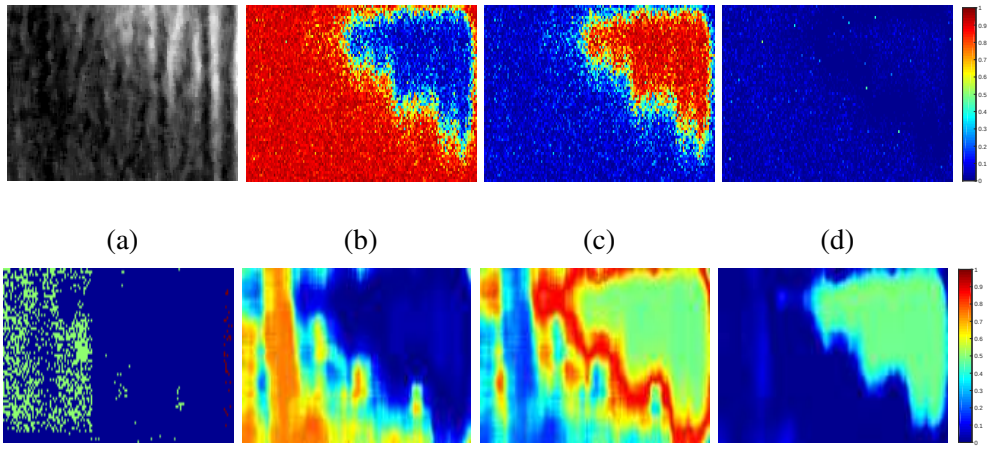


(a) SAS Image 1 (b) PM-LDA "dark flat sand" (c) PM-LDA "sand ripple" (d) PM-LDA "bright flat sand"

(e) LDA (f) FCM: 1 (g) FCM: 2 (h) FCM: 3

Figure 4.2: Segmentation results of SAS Image 1 using PM-LDA, FCM and LDA. (a) SAS Image 1. (b) PM-LDA partial membership map in the "dark flat sand" topic. (c) PM-LDA partial membership map in the "sand ripple" topic. (d) PM-LDA partial membership map in the "bright flat sand" topic. In (b) - (d), the color indicates the degree of membership of a visual word in a topic. (e) LDA result where color indicates topic label. (f) FCM partial membership map in the first cluster. (g) FCM partial membership map in the second cluster. (h) FCM partial membership map in the third cluster. In (f) - (h), the color indicates the degree of membership of a visual word in a cluster.

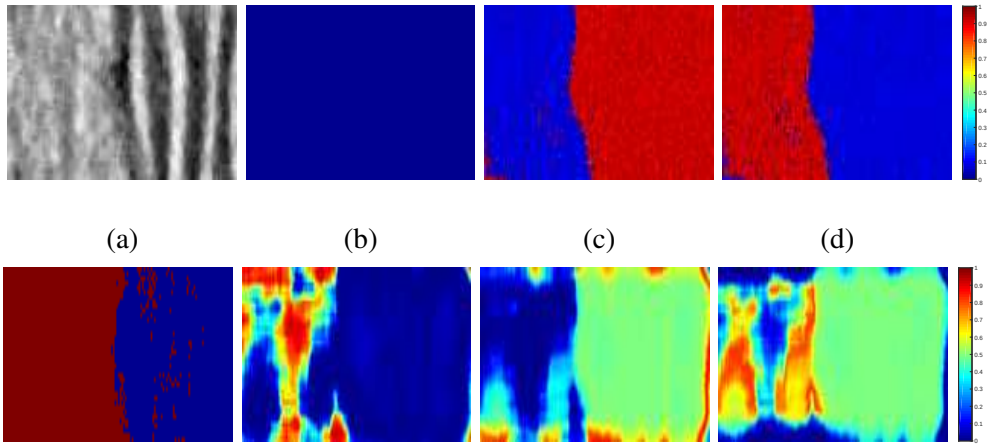
For LDA, the labeling confidence map is also computed by normalizing the label posterior density given the observations. Figure 4.6-4.9 shows such normalized posterior distri-



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 4.3: Segmentation results of SAS Image 2 using PM-LDA, FCM and LDA. Subfigure captions follow those in Figure 4.2.

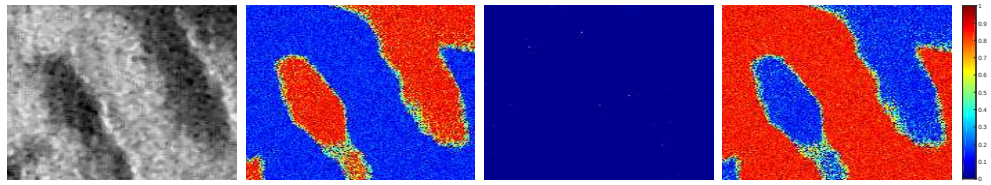


(a) (b) (c) (d)

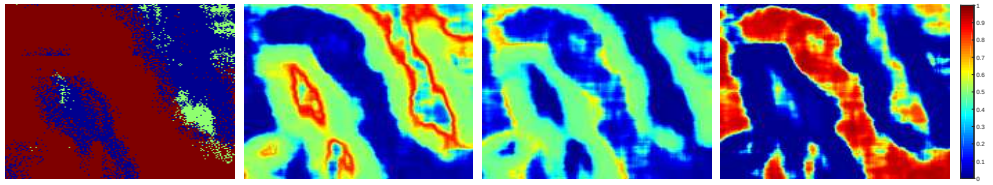
(e) (f) (g) (h)

Figure 4.4: Segmentation results of SAS Image 3 using PM-LDA, FCM and LDA. Subfigure captions follow those in Figure 4.2.

bution maps for a sonar image. Compared to results of PM-LDA and FCM, these are quite binary.



(a) (b) (c) (d)



(e) (f) (g) (h)

Figure 4.5: Segmentation results of SAS Image 4 using PM-LDA, FCM and LDA. Subfigure captions follow those in Figure 4.2.

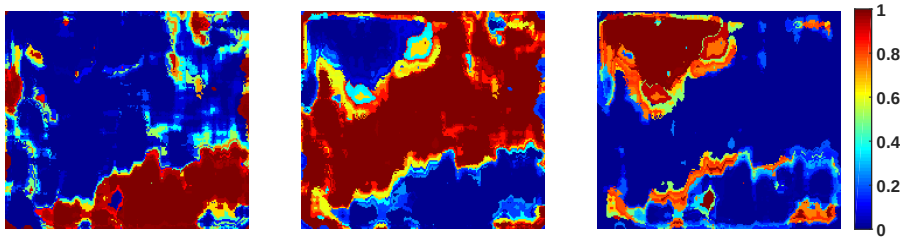


Figure 4.6: Normalized LDA Posterior of SAS Image 1 for Topics 1-3.

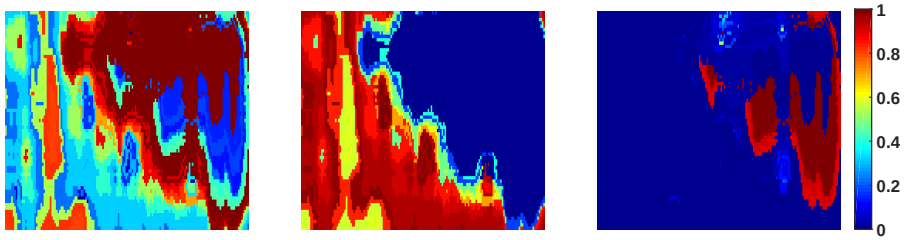


Figure 4.7: Normalized LDA Posterior of SAS Image 2 for Topics 1-3.

4.2.2 Using Superpixel as Document

The experiment are further extended to the complete SONAR imagery. Six complete SONAR images are used and named as HF_00, HF_01,..., HF_05, respectively. Each

Figure 4.8: Normalized LDA Posterior of SAS Image 3 for Topics 1-3.

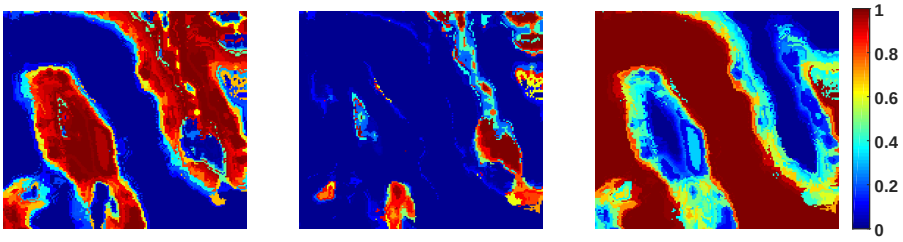


Figure 4.9: Normalized LDA Posterior of SAS Image 4 for Topics 1-3.

SONAR imagery is segmented into multiple superpixels (i.e., Figure 4.10) [56] and each superpixel is considered as a document. Besides the mean and entropy features used in the above experiment, a filter response is also used as the third feature in this experiment. The filter is built based on the sand ripple characterization algorithm proposed in [57]. It is hypothesized that each superpixel is a rippled region with certain ripple frequency, f_{ripple} (number of ripples per meter). The sand ripple characterization algorithm is applied to each superpixel to estimate its ripple frequency. As the range resolution of the sonar imagery is $0.025m$, a complete ripple length L can be computed as $40/f_{\text{ripple}}$. To capture the ripple repeating pattern, a filter is built as $[-\mathbf{1}, \mathbf{0}, \mathbf{1}, -\mathbf{1}, \mathbf{0}, \mathbf{1}]$, where $\mathbf{1}$ and $\mathbf{0}$ are matrices with height of 11 and width of $\lceil L/3 \rceil$. The filter is applied to the corresponding superpixel and the filter response is used as the third feature for that superpixel. Non-rippled superpixels have low filter responses and ripple superpixels have high filter response. The parameter estimation is run on each SONAR imagery individually. The parameter settings for the six

SONAR images are the same. The hyper-parameter λ and $\boldsymbol{\alpha}$ is set to be 0.1 and $\mathbf{0.1}_K$. For SONAR imagery HF_00 and HF_01, the topic number $K = 3$ and for SONAR imagery HF_02 - HF_05, the topic number $K = 2$. FCM is also applied in this experiment for comparison. The *fuzzifier* m is varied to be 1.1 to 3.0 with increment 0.1. The segmentation results of PM-LDA and FCM are shown in Figure 4.11 - 4.16. Figure 4.11 shows that PM-LDA and FCM achieve similar segmentation result on SONAR imagery HF_00. Both of them are capable to learn the "dark flat sand", the "sea grass", and the "sand ripple" topics. On SONAR imagery HF_01, as shown in Figure 4.12, PM-LDA performs better than FCM. FCM fails to cluster the "sand ripple" region an independent topic, which is roughly evenly split into the learned topic 2 and topic 3. For SONAR imagery HF_02 to HF_05, FCM fails to learn the "shadow" topic since the "shadow" region contributes a small part of the complete imagery and FCM prefers data points that are equally distributed into K topics. PM-LDA achieves better overall performance than FCM on SONAR imagery.

A further experiment is run on two SONAR imagery HF_00 and HF_01 using the same parameter setting. The segmentation results are shown in Figure 4.17. It can be seen that PM-LDA is still capable to learn the "dark flat sand", the "sea grass", and the "sand ripple" topics.

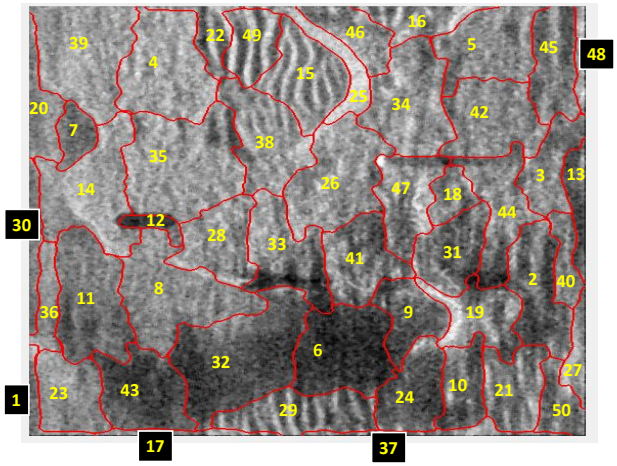
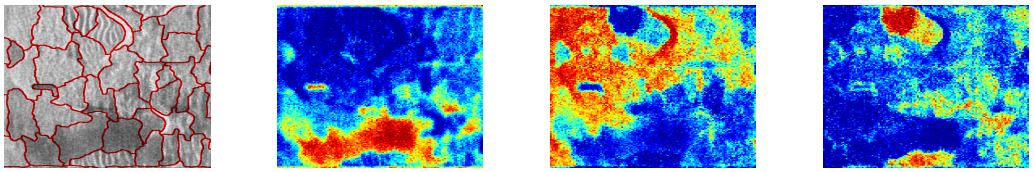
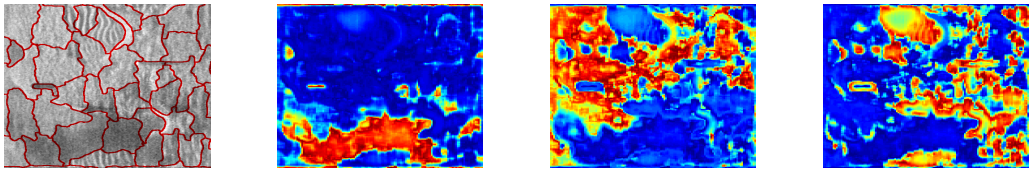


Figure 4.10: Superpixels in SONAR imagery HF_00.

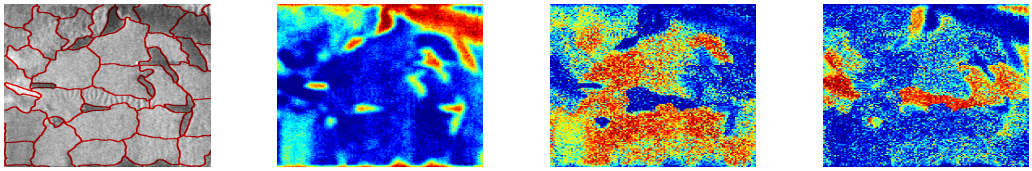


(a) SONAR imagery HF_00 (b) PM-LDA: Topic 1 (c) PM-LDA: Topic 2 (d) PM-LDA: Topic 3

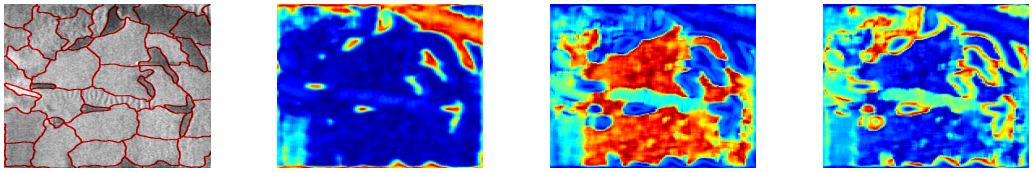


(e) SONAR imagery HF_00 (f) FCM: Topic 1 (g) FCM: Topic 2 (h) FCM: Topic 3

Figure 4.11: Segmentation result of SONAR Imagery HF_00 using PM-LDA and FCM. The *fuzzifier* $m = 2.0$. Topic 1 to 3 can be interpreted as "dark flat sand", "sea grass", and "sand ripple", respectively.

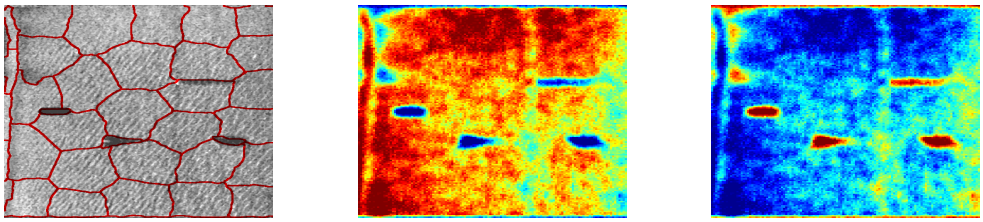


(a) SONAR imagery HF_01 (b) PM-LDA: Topic 1 (c) PM-LDA: Topic 2 (d) PM-LDA: Topic 3

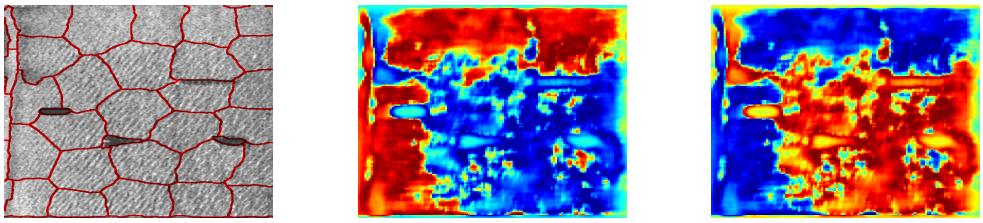


(e) SONAR imagery HF_01 (f) FCM: Topic 1 (g) FCM: Topic 2 (h) FCM: Topic 3

Figure 4.12: Segmentation result of SONAR Imagery HF_01 using PM-LDA and FCM. The *fuzzifier* $m = 2.3$. Topic 1 to 3 can be interpreted as "dark flat sand", "sea grass", and "sand ripple", respectively.

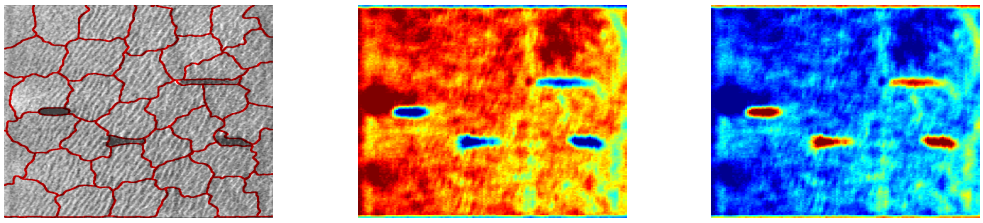


(a) SONAR imagery HF_02 (b) PM-LDA: Topic 1 (c) PM-LDA: Topic 2

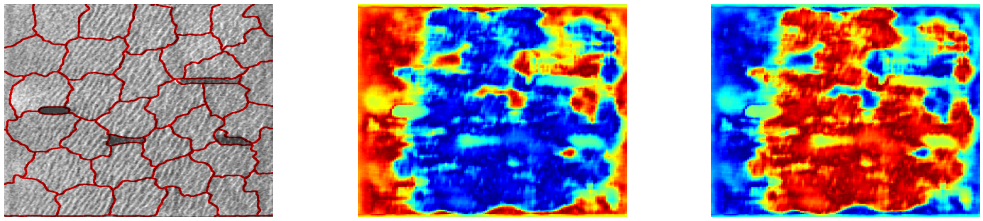


(d) SONAR imagery HF_02 (e) FCM: Topic 1 (f) FCM: Topic 2

Figure 4.13: Segmentation result of SONAR Imagery HF_02 using PM-LDA and FCM. The *fuzzifier* $m = 1.9$. Topic 1 and 2 can be interpreted as "shadow" and "sand ripple", respectively.

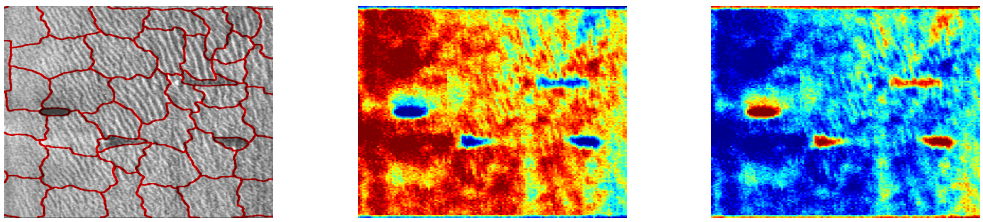


(a) SONAR imagery HF_03 (b) PM-LDA: Topic 1 (c) PM-LDA: Topic 2

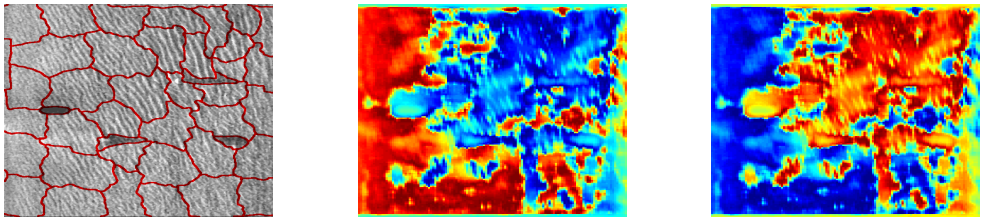


(d) SONAR imagery HF_03 (e) FCM: Topic 1 (f) FCM: Topic 2

Figure 4.14: Segmentation result of SONAR Imagery HF_03 using PM-LDA and FCM. The *fuzzifier* $m = 2.0$. Topic 1 and 2 can be interpreted as "shadow" and "sand ripple", respectively.

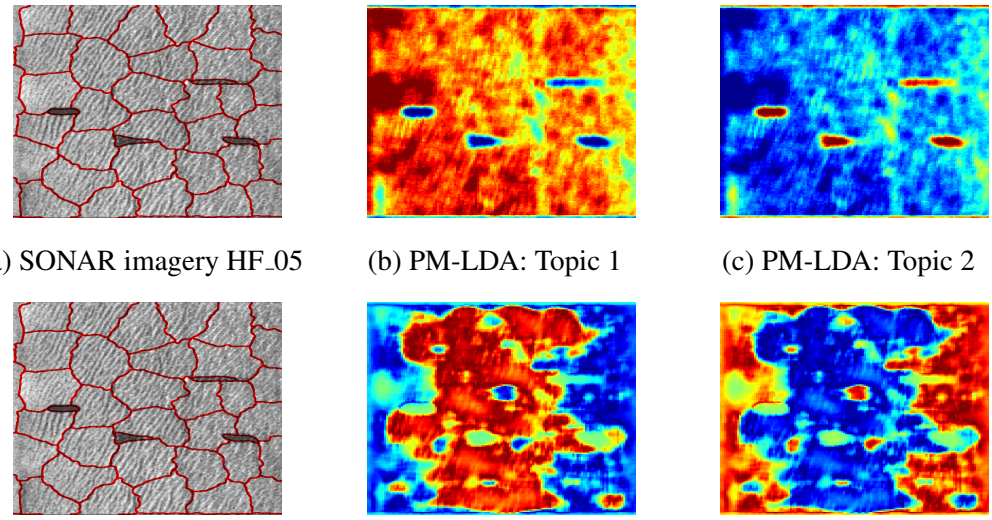


(a) SONAR imagery HF_04 (b) PM-LDA: Topic 1 (c) PM-LDA: Topic 2



(d) SONAR imagery HF_04 (e) FCM: Topic 1 (f) FCM: Topic 2

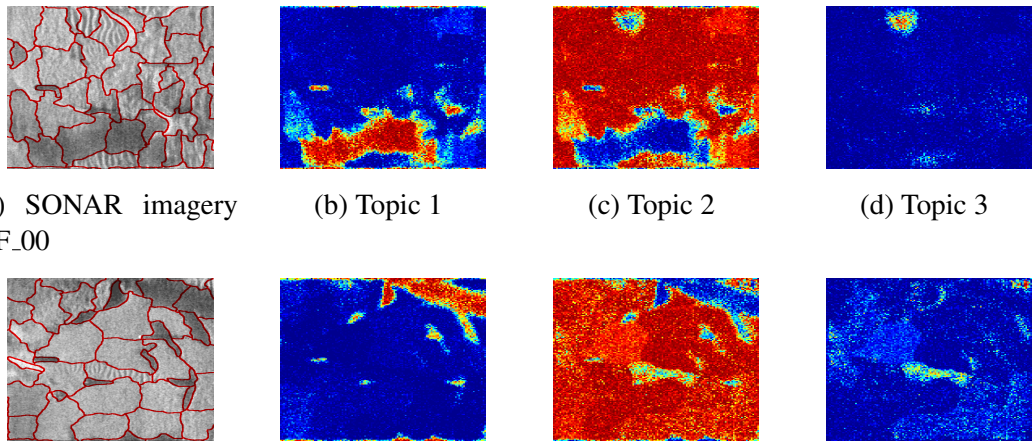
Figure 4.15: Segmentation result of SONAR Imagery HF_04 using PM-LDA and FCM. The *fuzzifier* $m = 1.9$. Topic 1 and 2 can be interpreted as "shadow" and "sand ripple", respectively.



(a) SONAR imagery HF_05 (b) PM-LDA: Topic 1 (c) PM-LDA: Topic 2

(d) SONAR imagery HF_05 (e) FCM: Topic 1 (f) FCM: Topic 2

Figure 4.16: Segmentation result of SONAR Imagery HF_05 using PM-LDA and FCM. The *fuzzifier* $m = 1.9$. Topic 1 and 2 can be interpreted as "shadow" and "sand ripple", respectively.



(a) SONAR imagery HF_00 (b) Topic 1 (c) Topic 2 (d) Topic 3

(e) SONAR imagery HF_01 (f) Topic 1 (g) Topic 2 (h) Topic 3

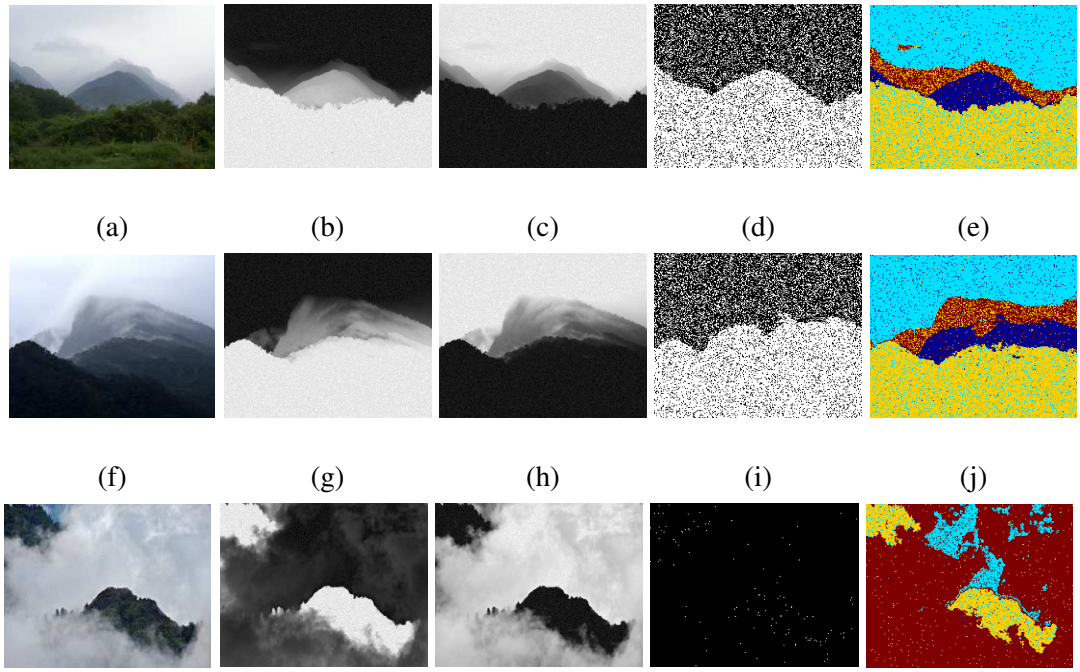
Figure 4.17: Segmentation results using two SONAR Imagery, HF_00 and HF_01. Topic 1 to 3 can be interpreted as "dark flat sand", "sea grass", and "sand ripple", respectively.

4.3 Experiments on Visual Natural Imagery

Fog-Mountain Dataset For the second set of experiments, a Fog-Mountain dataset is built to show the capacity of PM-LDA for partial membership segmentation in visual natural imagery. We selected 16 fog and mountain themed images from Flickr (with the necessary permissions). The RGB color values are used as the feature vectors for each visual word. For PM-LDA, the number of topics is set to be 2. For LDA, the number of topics is varied to be both 2 and 4. Experimental results are shown in Figure 4.18. The second and third column are the segmentation results of PM-LDA. The fourth column is the LDA results with 2 topics. Comparing the second and fourth column in Figure 4.18, it can be seen that (1) PM-LDA can generate continuous partial membership according to the extent to which the mountain is veiled by fog. The segmentation result (partial membership map) tells us how a topic gradually changes to another one. LDA just generates 0-1 membership; (2) Segmentation results of PM-LDA are very smooth while LDA segmentation results are quite noisy since the spatial contiguity is not guaranteed in LDA; (3) and as LDA tends to group co-occurring visual words into one topic, many visual words in "mountain" topic are mislabeled as "fog" topic and vice versa. In the last image where fog is dominating the image, almost all the "mountain" visual words are labeled as "fog" using LDA.

For a more fair comparison, the number of topics in LDA is increased to allow LDA to capture some of the regions of transition as new separate topics. As shown in the fifth column in Figure 4.18, when the number of topics is 4, LDA does capture some of the transition regions as new topics, i.e.,, the red topic and the dark blue topic. However, without looking at the original image, it is impossible to tell which topics correspond to the pure regions and which topics correspond to the transition regions. Even identifying these

regions, LDA can not localize the transition regions and can not provide the same transition information as PM-LDA.



(a) (b) (c) (d) (e)

(f) (g) (h) (i) (j)

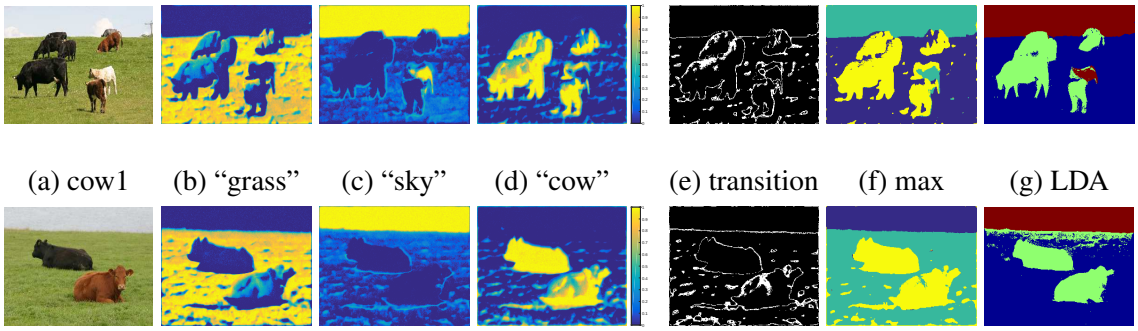
(k) (l) (m) (n) (o)

Figure 4.18: Examples of segmentation results on Fog-Mountain dataset. (a), (f), and (k) are the original images [1] [2] [3]. (b), (g), and (l) are the PM-LDA partial membership maps to the "mountain" topic. (c), (h), and (m) are the PM-LDA partial membership maps to the "fog" topic. (d), (i), and (n) are the LDA results with 2 topics. (e), (j), and (o) are the LDA results with 4 topics where color indicates the topic.

Microsoft Research Cambridge data set version one (MSRCv1) The MSRCv1 database consists of 240, 213×320 pixel images. We built a "cow"subset from this database by selecting images covering "grass", "cow", or "sky" topics. The "cow" subset has 44 images in total. We use the local descriptors proposed in [58], the output of a set of filter

[1] Image can be found at: https://www.flickr.com/photos/onbangladesh/858115496/
[2] Image can be found at: https://www.flickr.com/photos/jhoyos/3218520810/
[3] Image can be found at: https://www.flickr.com/photos/coloneljohnbritt/9245802086/

bank responses made of 3 Gaussians, 4 Laplacian of Gaussians (LoG) and 4 first order derivatives of Gaussians as the feature vectors. The filter window size is 15×15. In this experiment, instead of using each image as a document, normalized cuts method is applied to get 40 segments from each image and treat a segment as a document. The topic number is set to be 3, and for LDA, the filter bank output is densely sampled and a dictionary of size 200 is built. The ROC curve of PM-LDA is shown in Figure 4.20. The red start indicates the LDA result. Some examples of results using PM-LDA and LDA are shown in Figure 4.19. Subfigures (b)-(d) show the partial membership maps in each of the three topics. We highlight the transition regions by pulling out the pixels with at least one membership value in range $[0.4, 0.6]$. As shown in (e), these partial membership values mostly occur at the boundary between two topics. Thus, PM-LDA is able to identify when the feature vector contains information from multiple topics (as the feature vector is being computed over a window that contains more than one topic). This is a powerful result showing the effectiveness of PM-LDA to provide semantic image understanding. For comparison with LDA, the segmentation result of PM-LDA is modified by assigning each visual word to the topic with the largest membership. As shown in (f) and (g), PM-LDA can achieve similar results to LDA. So on these images with crisp boundaries, PM-LDA can generate binary membership values, and learn the three semantic topics comparable to LDA. Thus, PM-LDA also is effective for use in crisp labeling problems.



(a) cow1 (b) “grass” (c) “sky” (d) “cow” (e) transition (f) max (g) LDA

(a) cow2 (b) “grass” (c) “sky” (d) “cow” (e) transition (f) max (g) LDA

Figure 4.19: Example of PM-LDA and LDA results. (a) Original image. (b)-(d) Results of PM-LDA, the partial membership maps in “grass”, “sky”, and “cow” topics, respectively. The color indicates the degree of membership in a topic. (e) Transition regions consisting of visual words with at lease one partial membership value in range $[0.4, 0.6]$ (f) Modified segmentation result of PM-LDA by assigning each visual word to the topic with the largest membership. The colors indicate topics. (g) Result of LDA. The colors indicate topics.

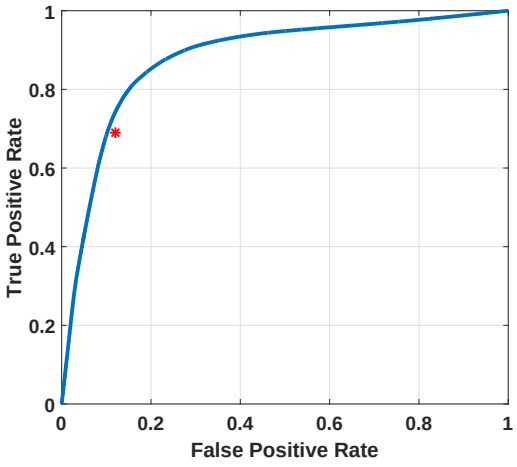


Figure 4.20: ROC curve of PM-LDA. The red star represents the LDA result.

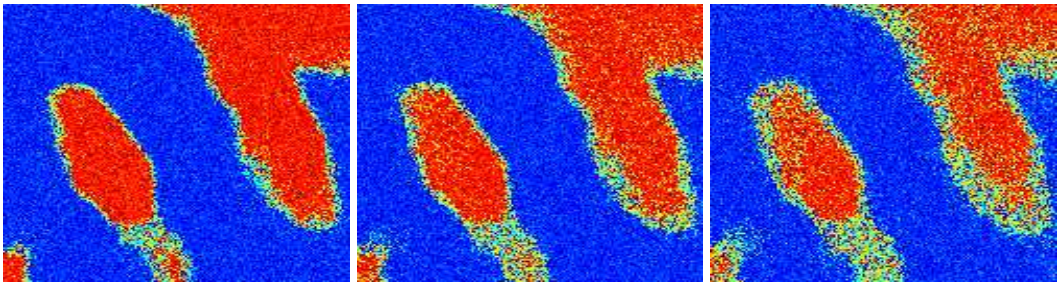
4.4 Experiments on Scaling factor λ and s

As discussed, the parameter λ determines the expected degree of mixing. This translates to the width of the transition regions in a SAS image. A large λ (small s) corresponds to a narrow transition region.

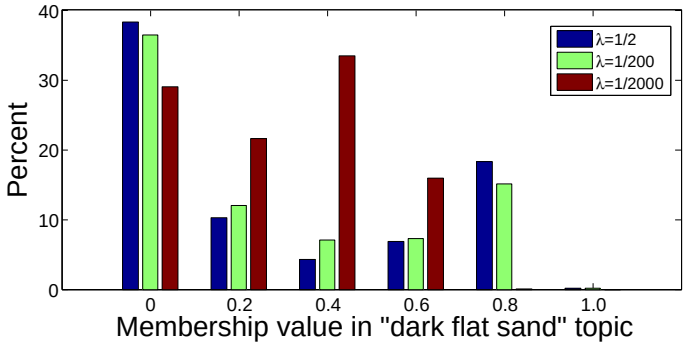
4.4.1 Effect of λ in Inference Procedure

This experiment investigated the effect of λ by estimating the memberships with fixed topics learned in the above experiment. λ is varied to be $1/2$, $1/200$, and $1/2000$. Segmentation results on Image 4 with different λ values are shown in Figure 4.21. Figure 4.21a - 4.21c show that as λ decreases, the estimated transition region becomes wider. Figure 4.21d shows the histogram of membership values in "dark flat sand" topic averaged over 10 runs. As λ decreases, the average percentage of membership value in range $[0.4, 0.6]$ becomes larger (i.e., there is more mixing).

As randomly drawing a scaling factor from an exponential distribution parameterized by λ can not guarantee an expected scaling factor. To better show the effect of scaling factor, the scaling factor are manually set. Another experiment is conducted to show the effect of s by estimating the memberships with fixed topics learned using SONAR Imagery HF_00 and HF_01. The scaling factor s is set to be 1, 5, and 10. The estimated membership maps in the learned Topic 1 to 3 for different s values are shown in Figure 4.22. As shown in Figure 4.22, the estimated transition region becomes wider as the scaling factor s increases.

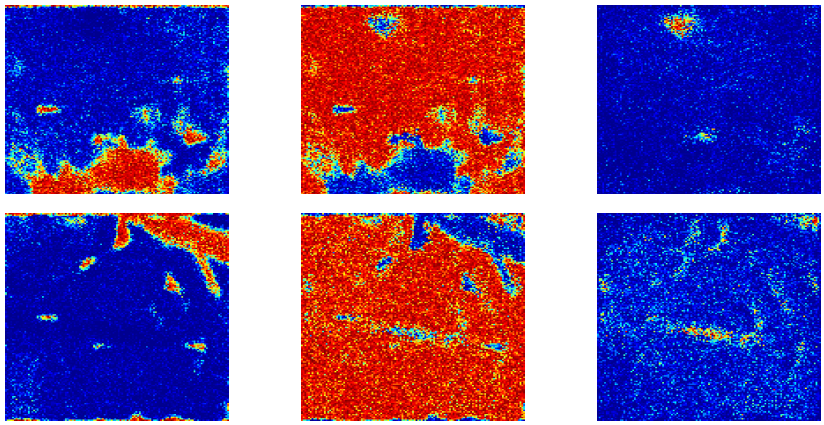


(a) $\lambda = 1/2$ (b) $\lambda = 1/200$ (c) $\lambda = 1/2000$

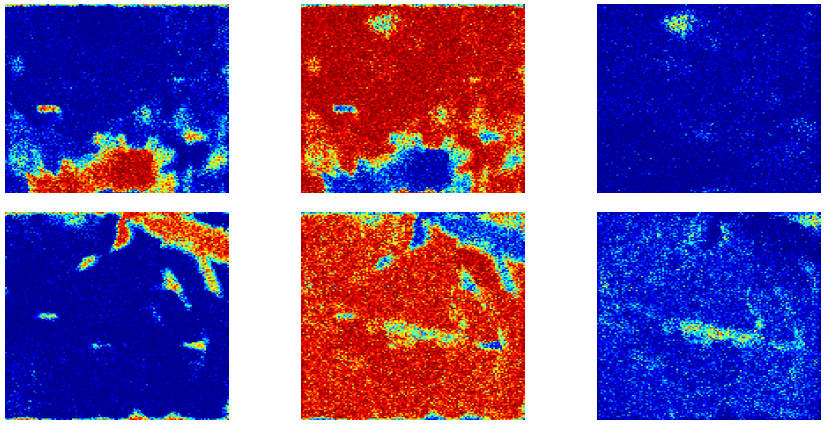


(d)

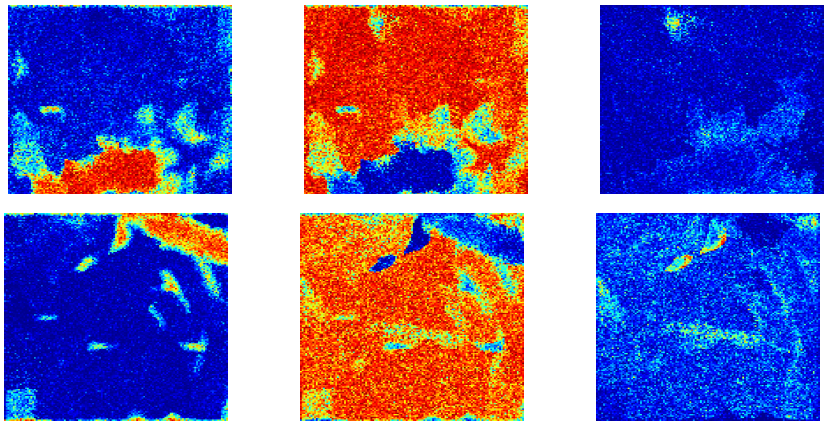
Figure 4.21: Partial membership map in “dark flat sand” topic for Image 4 with varying λ.



(a) $s = 1$, Topic 1 (b) $s = 1$, Topic 2 (c) $s = 1$, Topic 3



(d) $s = 5$, Topic 1 (e) $s = 5$, Topic 2 (f) $s = 5$, Topic 3



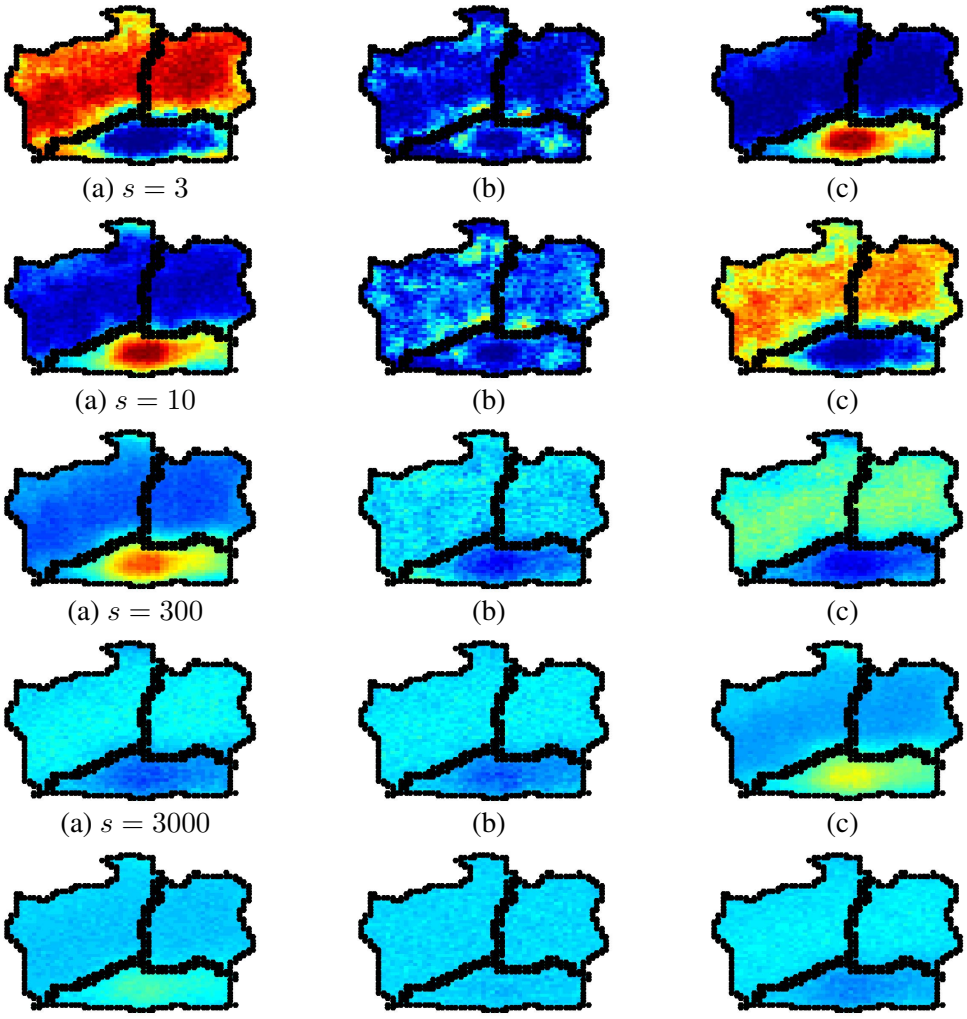
(g) $s = 10$, Topic 1 (h) $s = 10$, Topic 2 (l) $s = 10$, Topic 3

Figure 4.22: Partial membership maps in Topic 1-3 with varying s.

4.4.2 Effect of s in Parameter Estimation Procedure

The experiment is further extended to study the effect of s in the parameter estimation procedure. As described in the PM-LDA generative process, a membership vector $\mathbf{z}$ is drawn from Dir($s\boldsymbol{\pi}$). The scaling factor s controls the similarity of memberships to the topic proportion. For example, with big s such that $s\boldsymbol{\pi}_k > 1$, the memberships are preferred to be densely distributed around the topic proportion $\boldsymbol{\pi}$. With small s such that $s\boldsymbol{\pi}_k < 1$, the memberships are preferred to be crisp and the vast majority of the memberships will be concentrated at a few of crisp vectors (i.e., $[1, 0, 0]$, $[0, 1, 0]$ and $[0, 0, 1]$ when $K = 3$). And when $s\boldsymbol{\pi}_k = 1$, the memberships are preferred to be uniformly distributed. In this experiment, three superpixels labeled as 6, 29 and 32 in Figure 4.10 are extracted from a SONAR imagery and used as three documents. The topic proportion is fixed to be $\boldsymbol{\pi} = [1, 1, 1]/3$ and the scaling factor s is varied to be $3, 10, 300, 3000, 30000$. The membership estimation results are shown in Figure 4.23. As the scaling factor s increases, the partial memberships gradually approach the topic proportion $[1, 1, 1]/3$.

The experiment is further extended to the whole SONAR imagery. The scaling factor s is varied to be 5, 20 and 30 and the topic proportion $\boldsymbol{\pi}$ is learned for each superpixel (document). The proposal distribution for $\boldsymbol{\pi}$ is Dir($\mathbf{0.1}$) and the topic number $K = 3$. Figure 4.24 shows that as the scaling factor s increases, the memberships within a superpixel become more and more similar to each other. The memberships map in each superpixel becomes more smoother.



(a) $s = 3$ (b) (c)

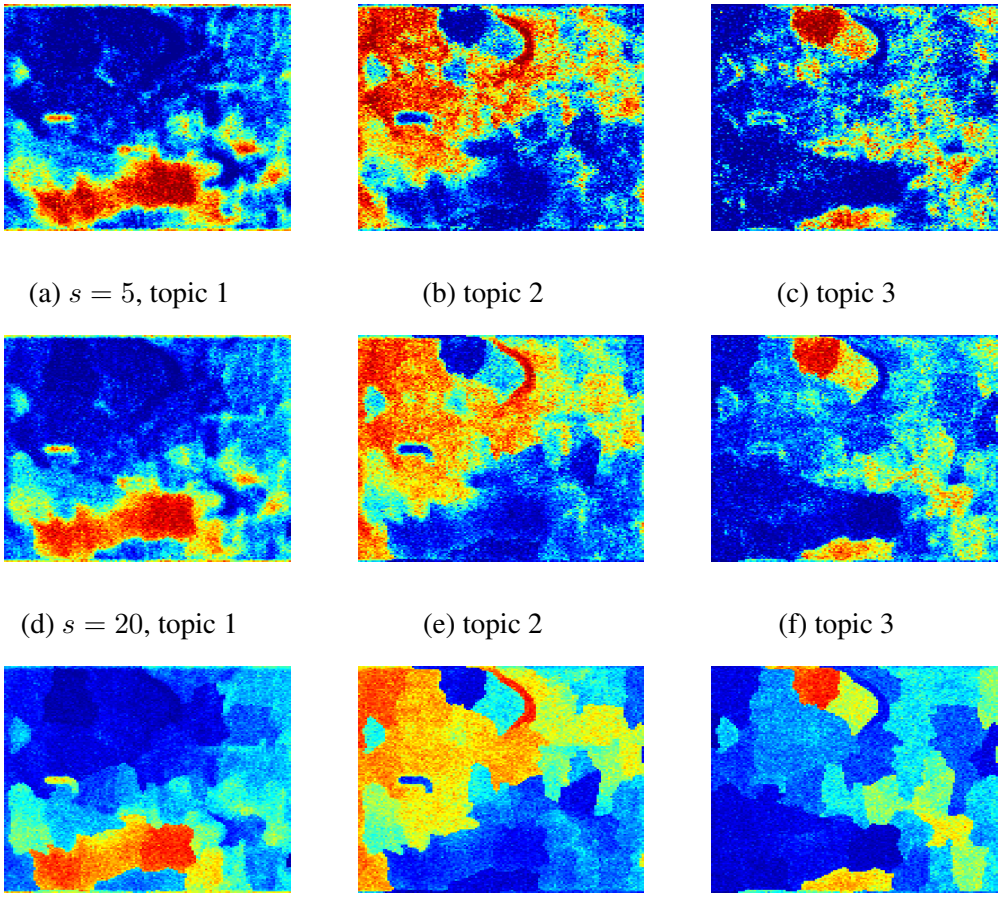
(a) $s = 10$ (b) (c)

(a) $s = 300$ (b) (c)

(a) $s = 3000$ (b) (c)

(a) $s = 30000$ (b) (c)

Figure 4.23: Partial membership maps with varying s. Each row shows the estimated membership maps of the three estimated topics. The black contour indicates the superpixel boundary.



(a) $s = 5$, topic 1 (b) topic 2 (c) topic 3

(d) $s = 20$, topic 1 (e) topic 2 (f) topic 3

(g) $s = 30$, topic 1 (h) topic 2 (i) topic 3

Figure 4.24: Estimated Memberships in topic 1-3 with varying scaling factor s. The three columns represent the memberships in topic 1 to 3, respectively.

Chapter 5

Summary and Future Work

In this work, the PM-LDA model is introduced for soft image segmentation. PM-LDA improves upon the LDA model by introducing a partial membership rather than requiring a single topic label for each word. Experimental results on three image datasets, SAS imagery dataset, Fog-Mountain dataset, and MSRCv1 dataset, demonstrate the capacity of PM-LDA model in both soft and crisp image segmentation.

Future work will include developing a more efficient parameter estimation approach, e.g., collapsed Gibbs sampler, variational message passing, to accelerate the parameter estimation procedure. Another research direction is accelerating PM-LDA model with state-of-the-art hardware, e.g. Nvidia GPU, Intel Xeon Phi, which have been widely used in application acceleration [59–61]. For topic model, other exponential family distribution, e.g., Rayleigh distribution, will be investigated. Furthermore, the topic distribution can be extended beyond exponential family. For example, the topic distribution can be modeled as a mixture of Gaussian to use the co-occurrence to help clustering. For sonar imagery application, more SAS imagery features, such as features extracted from the reconstructed

bathymetry map, will be explored. For PM-LDA, more experiments will be conducted to examine how to determine topic numbers, set parameters, and balance ratio of data points from different topics.

Appendix A

Appendix 1 - Sand Ripple Characterization using an Extend Synthetic Aperture Sonar Model and Parallel Sampling Method

Accurate estimates of the sand ripple frequency, amplitude, and orientation characteristics can be ultimately used within a number of applications of SAS imagery including detection and classification of targets in a scene [54, 55], seabed segmentation and classification [52, 53], and others. This work presents a method for estimating sand ripple frequency, amplitude, and orientation values from a single SAS image as well as from sets of SAS imagery over an area using a hierarchical Bayesian framework and a known sensing geometry. This is accomplished through the development of an extended model for sand ripple characterization and a Metropolis-within-Gibbs sampler to estimate sand ripple frequency, amplitude, and orientation characteristics for multi-aspect high-frequency side-look sonar data. Results are presented on synthetic and measured SAS imagery that indicate the ability

of the proposed method to estimate desired sand ripple characteristics.

A.1 Sand Ripple Model

A stochastic model for high frequency and narrow angle Synthetic Aperture Sonar (SAS) imagery collected over a sand ripple field, originally proposed by Lyons, et al. [50], represents the image as a product between a process governing the speckle in the imagery, $Z(r, x)$, and a process governing the amplitude of a pixel's scattering strength, $a(r, x)$,

$$Y(r, x) = Z(r, x)a(r, x) \tag{A.1}$$

where r and x are the down-range and cross-range image dimensions, respectively. In this model, considering only the down-range dimension, r, the amplitude of a pixel's scattering strength is approximated by

$$a(r) \simeq \sigma_s(\theta_0) + \left[\sigma_s\left(\theta_{max}\right) - \sigma_s\left(\theta_{min}\right)\right] \frac{g(r) - \bar{g}}{g_{max} - g_{min}}, \tag{A.2}$$

where $\sigma_s(\theta)$ is the scattering cross section at the grazing angle θ, $g(r)$ is the slope of seafloor at range r, and θ_0 is the average grazing angle. The max and min values are the extremal values of θ and g around the local grazing angle, and $\bar{g}$ is the average slope. The scattering cross section is a common statistical model describing seafloor scattering where acoustic waves are randomly scattered by irregularities on the sea floor. It measures the redistributed acoustic energy caused by the scattering process [62]. Eqn. (A.2) is graphically depicted in Figure A.1.

In our work, the speckle noise $Z(r, x)$ which is assumed to follow a K-distribution with

a large shape parameter is approximated using a Rayleigh distribution. Since the Rayleigh distribution has only one scale parameter, s, tuning this parameter can only change the noise scale, while the shape of the probability mass function of the noise remains unchanged. An example of a synthetic ripple field generated using this model is shown in Figure A.2.

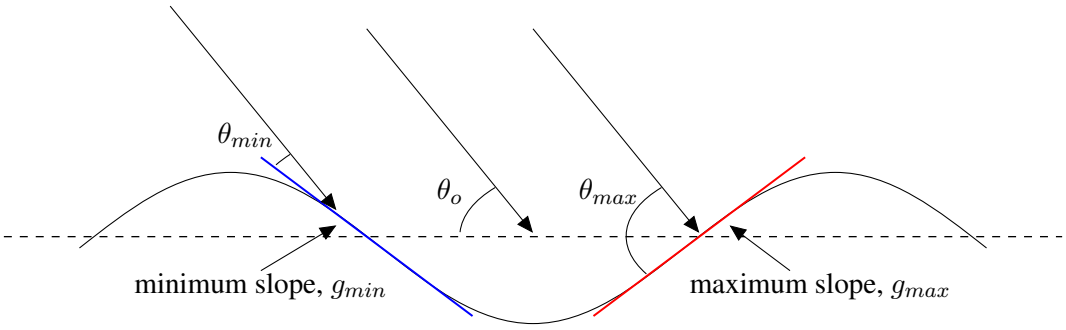


Figure A.1: Graphical depiction of the original sand ripple scattering model from [50] .

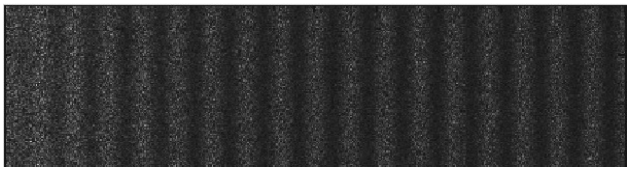


Figure A.2: Synthetic ripple field generated using the model in [50] with a sine wave bathymetry profile. In this figure, $\sigma_s(\theta)$ was simulated using a Normal distribution function centered at $\theta = \frac{\pi}{2}$. The speckle term, $Z(x, y)$, for this image was generated using a Rayleigh distributed speckle with a parameter value of $s = 1 \times 10^2$.

A.2 Occluded Sand Ripple Model

In this work, we further develop Lyons' stochastic model by refining the scattering cross-section function based on (A.2) to explicitly incorporate the occlusion from preceding peaks in the sand ripple field. Then, using this expanded model, the parameters governing the bathymetry profile (i.e. the frequency, amplitude, and orientation of the peaks in

the sand ripple height field) are estimated from SAS imagery using a hierarchical Bayesian framework in conjunction with a Metropolis-within-Gibbs sampling algorithm.

The expanded sand ripple field model is defined in Section A.2. Section A.3 describes the proposed hierarchical Bayesian framework and the Metropolis-within-Gibbs sampling algorithm used to estimate the desired parameters from both single passes over an area and using multiple aspects and passes over an area at a variety of ranges. Section A.4 includes experimental results on simulated and measured SAS imagery. Section A.5 summarizes the research and discusses future work.

The proposed method estimates bathymetry information (i.e., sand ripple field height) by leveraging occlusions in the imagery caused by preceding peaks in the sand ripple field. To accomplish this, the model in (A.2) is expanded to explicitly include a scattering cross-section function that incorporates occlusion of scattering from the ripple trough by the preceding peaks. In this expanded model a function ψ_s replaces σ_s in (A.2),

$$a^*(r, A_H, f, \beta) \simeq \psi_s(\theta_0, r) + [\psi_s(\theta_{max}, r) - \psi_s(\theta_{min}, r)] \frac{g(r) - \bar{g}}{g_{max} - g_{min}}, \tag{A.3}$$

with

$$\psi_s(\theta, r) \triangleq \begin{cases} 0 & \text{if } 2 \cdot A_H - \frac{A_S \cdot g(r)}{r - g(r)} \geq h(r) \\ \sigma_s(\theta) & otherwise \end{cases}, \tag{A.4}$$

where A_H is the ripple amplitude, f is the ripple frequency, β is the ripple orientation, A_S is the height of the SAS array from the sea floor, and $h(r)$ is the ripple height at range r. Note that an estimate of A_S is measured and provided by the SAS sensor for each downrange location. However, the A_S value is dependent on the sea-floor characteristics below the sensor. For example, if the SAS array is located above sand ripple, then the A_S value will depend on whether the measurement is collected over a peak or trough of the sand ripple

field.

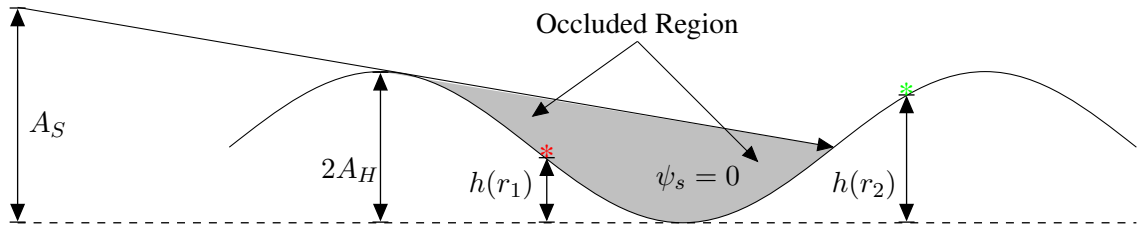


Figure A.3: The expanded model that accounts for shallow grazing angles at long ranges by including the case of occlusion of the ripple trough. The red asterisk is obscured by the preceding peak, while the green asterisk is visible by the sonar.

(a) $\beta = 0 \quad (0°)$ (b) $\beta = \pi/9 \quad (20°)$

Figure A.4: Synthetic ripple field displaying occluded pixels in the trough at long ranges. These shadowed or occluded pixels are typically seen in real imagery. The speckle term, $Z(x, y)$, for this image was generated using a Rayleigh distributed speckle with a parameter value of $s = 1 \times 10^2$.

Given the multiplicative noise model in the sand ripple model in (1), noise is multiplied to the scattering strength $a^*(r, x)$. Thus, given a response of zero in the occluded regions as shown in (A.4) and given that the input imagery are normalized during preprocessing, the speckle noise scale level should have no significant impact on the simulated SAS imagery and our proposed estimation method. However, noise due to multipath effects may cause a non-zero response in the occluded regions in real measured imagery. The impact of the noise is dependent on the ability to accurately locate the occluded regions. For imagery with noise that causes a non-zero response in the region of occlusion, a preprocessing step in which the occluded regions are detected and set to zero would allow for application of the proposed method without any change in performance.

A.2.1 Ripple Height Field

Sand ripples can be approximated by sinusoidal shapes [50]. Assume that the along-crest orientation is perpendicular to the down range direction, then the ripple height at (r, x) can be defined as

$$h(r, x) \simeq A_H \sin(2\pi f r + b) + A_H, \tag{A.5}$$

where f and b are the sand ripple frequency and phase values, respectively. In practice, since the along-crest orientation of the sand ripple is rarely parallel to the down-range direction, we also introduce into Equation (A.5) the ripple orientation β,

$$h(r, x) \simeq A_H \sin(2\pi f \cos(\beta) r - 2\pi f \sin(\beta) x + b) + A_H, \tag{A.6}$$

where β is the angle between the ripple along-crest orientation and the down-range axis. For each fixed x, the ripple field height at range r, can be simplified to

$$h(r) \simeq A_H \sin(2\pi f \cos(\beta) r + b') + A_H, \tag{A.7}$$

where $b' = -2\pi f \sin(\beta) x + b$. So for each x, the ripple height $h(r)$ is still a sine curve with identical amplitude A_H and frequency $f \cos(\beta)$ but with varying phase b'. The ripple slope field at range r can then be computed as

$$h'(r) = g(r) \simeq 2\pi f \cos(\beta) A_H \cos(2\pi f \cos(\beta) r + b'). \tag{A.8}$$

In this model, we estimate the ripple height field (i.e., bathymetry) without taking into account any overall sea floor slope. In our future work, we will investigate this factor and incorporate it into Equation (A.6)).

A graphical illustration of the role the additional variables play is depicted in Figure A.3. Synthetic ripple fields with different orientations are generated from this model and shown in Figure A.4. The synthetic ripple field shown in Figure A.4 has larger areas of occlusion at longer ranges as desired. Given this extended model, the relative width of occluded regions provides the needed information to estimate the sand ripple amplitude.

A.2.2 Models for Scattering Cross Section

Three models for the scattering cross section, σ_s, in (A.4) have been considered in this manuscript: Gaussian, $\sin^2$, and small slope approximation (SSA) models. The Gaussian and $\sin^2$ models are simple approximations that are functions only of the slope of the sand ripple field. The small slope approximation model also considers additional parameters such as sound speed in the water, surface roughness, and sediment grain size. Since normalized images are ultimately used in our work, these three models essentially provide us an estimate of the shape of the scattering curve, i.e., the relative scattering level. The absolute scattering level is lost through image normalization and a lack of the appropriate calibration. Figure A.5 shows the scattering cross section curves for the three models that are scaled to the range $[0, 1]$. The purpose of using all three of the Gaussian, sin^2, and SSA models in this work is to show that our proposed estimation method can be applied to SAS imagery when modeled by a variety of scattering cross section models.

Gaussian Model The first scattering cross section model considered is the Gaussian model. In this case, the scattering cross section is approximated as

$$\sigma_{s,Gauss}(\theta_r) = \mathcal{N}\left(\theta_r \middle| \frac{\pi}{2}, \sigma^2\right) \tag{A.9}$$

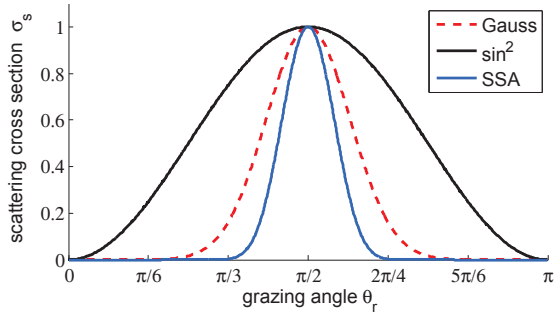


Figure A.5: The normalized scattering cross section curves of Gaussian, sin^2, and SSA models. For the Gaussian model, $\sigma_{s,Gauss}(\theta_r) = \mathcal{N}\left(\theta_r \left| \frac{\pi}{2}, (\frac{\pi}{6})^2 \right.\right)$, and for the sin^2 model, $\sigma_{s,\sin^2}(\theta_r) = \frac{1}{\pi}\sin^2(\theta_r)$. For the SSA model, $a_\rho = 2.0$, and $a_p = 1.1 - 0.1j$. The three curves in the figure are the scattering cross sections when normalized to $[0, 1]$.

Thus, with this model, the largest response is given when $\theta_r = \frac{\pi}{2}$ since a grazing angle perpendicular to the ripple height field is expected to provide the largest return. In our implementation, the variance of the Gaussian used to model the scattering cross section is set to be $\sigma^2 = (\frac{\pi}{6})^2$. This variance value was determined based on visual comparisons between data simulated using this model and collected SAS imagery.

$\sin^2$ **Model** The $\sin^2$ model is defined as

$$\sigma_{s,\sin^2}(\theta_r) = \mu \sin^2(\theta_r). \tag{A.10}$$

This model is widely referred to as Lambert's Law [62]. For backscattering from a slightly rough surface (e.g., soft sediments), its validity is restricted to oblique incidence angles. On very rough surfaces (e.g., rocky seabed) it is valid over the entire angular domain. The value of μ is often taken to be the empirical Mackenzie coefficient, which is about 0.002 (-27dB) [63]. If there is no penetration or other losses at the bottom, μ will be -5dB [64].

In our implementation, since the simulated images are normalized, the choice of μ has no impact on the final simulated image. We simply set μ to be $\frac{1}{\pi}$ (-5dB) .

SSA Model As one of the physics-based models for the rough surface scattering problem, the SSA model [62] considers the physical parameters of the seafloor. It takes into account the acoustic frequency and certain roughness-related parameters such as sediment grain size and sediment-water density ratio. The roughness scattering cross-section in the SSA model is

$$\sigma_{s,SSA}(\theta_r) = \frac{k_w^4 |A_{ww}|^2}{2\pi \Delta K^2 \Delta k_z^2} I_k, \tag{A.11}$$

where k_w is the wavenumber in water, ΔK is the magnitude of the horizontal component of the difference of the scattered and incident wave vectors, and Δk_z is the magnitude of the vertical component of this difference. The Kirchhoff integral, I_k, given a high frequency SSA approximation to the seafloor with isotropic roughness statistics is approximated by

$$I_k = \frac{cot^2(\theta_r)}{R_s^2} e^{-\frac{cot^2(\theta_r)}{2R_s^2}}, \tag{A.12}$$

where R_s is called the RMS slope. The A_{ww} term is affected by the above-mentioned roughness-related parameters and also depends on the choice of wave theory used as discussed in [62, p. 332-376][65]. For a fluid model, a convenient expression for A_{ww} is defined as

$$A_{ww} = \frac{1}{2}[1 + V(\theta_r)]^2 G, \tag{A.13}$$

with

$$G = (1 - \frac{1}{a_\rho})[\cos^2(\theta_r) - B] - 1 + \frac{1}{a_p^2 \cdot a_\rho}, \tag{A.14}$$

where a_ρ is the sediment-water density ratio, a_p is the complex-ratio of sediment compressional wave speed to water sound speed, and $V(\theta_r)$ is the flat-interface reflection coefficient at θ_r. Values of these parameters depend on the grain size. Refer to [62, p. 340-341] for more details on the term B, and V.

A.3 Hierarchical Bayesian Framework and Metropolis-within-Gibbs Sampler

The proposed method estimates the sand ripple bathymetry by sampling the ripple frequency, amplitude and orientation, (f, A_H, β) using a Metropolis-within-Gibbs sampling algorithm [66], a Markov chain Monte Carlo (MCMC) method that samples from a probability distribution using a Markov chain with the desired distribution as its stationary distribution [67, 68]. The sample set (f, A_H, β) that minimizes the difference between the true image and the corresponding simulated image based on the extended model in (A.4) is selected as the final estimate. The Metropolis-within-Gibbs sampler allows for generating samples from a multivariate distribution. This approach provides the advantage of being able to estimate a full distribution of possible parameter values given a complex data likelihood and prior distribution while maintaining the convergence guarantees of an MCMC sampling approach. The Metropolis-within-Gibbs sampler was implemented to estimate frequency, amplitude, and orientation characteristics given the following hierarchical Bayesian framework.

A.3.1 Hierarchical Bayesian Framework

The data likelihood used in the proposed framework places a Gaussian distribution around an input SAS pixel value given its estimate with the expanded model of the amplitude of the pixel's scattering strength,

$$x_{i,r}|A_H, f, \beta \sim \mathcal{N}\left(x_{i,r}|a^*(r, A_H, f, \beta), \sigma_x^2\right), \tag{A.15}$$

where $x_{i,r}$ is the i^{th} pixel in the image at range r. In our work, the pixel values are assumed to be in the range $[0, 1]$. After computing a^*, we normalized it to the range $[0, 1]$. σ_x is set to be 1 so that the majority of the pixel values are still in the range $[0, 1]$. Then, the likelihood over a set of SAS image pixels, assuming independence, can be defined as shown in (A.16),

$$\mathbf{X}|A_H, f, \beta \sim \prod_{i=1}^{N} \mathcal{N}\left(x_{i,r}|a^*(r, A_H, f, \beta), \sigma_x^2\right), \tag{A.16}$$

where N is the total number of pixels in the SAS image under consideration and $\mathbf{X}$ is the set of these image pixels. By determining the A_H, f and β values that maximize the likelihood function in (A.16), the mean squared error between the estimate given the model in (A.3) and the input data, $\mathbf{X}$, is minimized.

In order to constrain the sand ripple characteristics to physically possible values and incorporate any prior information, truncated prior distributions are placed on the A_H, f and β values. For the first (or single) pass of the data over an area, the prior distributions are truncated uniform distributions,

$$A_H \sim \mathscr{U}(A_H|l_{A_H}, u_{A_H}), \tag{A.17}$$

$$f \sim \mathscr{U}(f|l_f, u_f), \tag{A.18}$$

and

$$\beta \sim \mathscr{U}(\beta|l_\beta, u_\beta). \tag{A.19}$$

The lower and upper truncation points, $l_{A_H}, u_{A_H}, l_f, u_f, l_\beta$ and u_β are set to constrain the ripple amplitude, frequency, and orientation to physically possible values. These values may also be assigned given additional oceanographic information such as sediment type, wave/current information, etc. For the subsequent passes, the prior distributions are truncated Gaussian prior distributions,

$$A_H \sim \mathscr{N}_t(A_H|\mu_{A_H}, \sigma_{A_H}, l_{A_H}, u_{A_H}), \tag{A.20}$$

$$f \sim \mathscr{N}_t(f|\mu_f, \sigma_f, l_f, u_f), \tag{A.21}$$

and

$$\beta \sim \mathscr{N}_t(\beta|\mu_\beta, \sigma_\beta, l_\beta, u_\beta), \tag{A.22}$$

with

$$\mathscr{N}_t(x|\mu, \sigma, l, u) = \begin{cases} C\mathscr{N}(x|\mu, \sigma) & \text{if } l \leq x \leq u \\ 0 & otherwise \end{cases}, \tag{A.23}$$

and

$$C = \frac{1}{\Phi(\frac{u-\mu}{\sigma}) - \Phi(\frac{l-\mu}{\sigma})}, \tag{A.24}$$

where C is the standard normalization for the truncated Gaussian distribution, and $\Phi(\cdot)$ is the cumulative distribution function of standard normal distribution. The prior parameter values $(\mu_{A_H}, \sigma_{A_H}, \mu_f, \sigma_f, \mu_\beta, \sigma_\beta)$ are set based on estimates given by previous passes over the sand ripple field under consideration.

A.3.2 Sampling Method

Metropolis-within-Gibbs sampler

This approach iteratively samples f, A_H and β using a Metropolis-within-Gibbs algorithm [66]. The sampling algorithm for estimating the values given a single pass of the data is summarized in the pseudo-code in Algorithm 6. In the following, each step of the sampling algorithm is described.

Algorithm 4 *A Metropolis-within-Gibbs sampler that samples sand ripple frequency, amplitude, and orientation from* $\Pi(f, A_H, \beta|\mathbf{X})$. *The method used for each step is described in the section shown in parentheses.*

1: Set parameter values and initialize $f^{(0)}$, $A_H^{(0)}$ and $\beta^{(0)}$ (Section A.3.2)
2: **for** $k \leftarrow 1$ to N **do**
3: Sample the ripple frequency, $f^{(k)} \sim \Pi(f|\mathbf{X}, A_H^{(k-1)}, \beta^{(k-1)})$ (Section A.3.2)
4: Sample the ripple amplitude, $A_H^{(k)} \sim \Pi(A_H|\mathbf{X}, f^{(k)}, \beta^{(k-1)})$ (Section A.3.2)
5: Sample the ripple orientation, $\beta^{(k)} \sim \Pi(\beta|\mathbf{X}, f^{(k)}, A_H^{(k)})$ (Section A.3.2)
6: **end for**
7: **return** $(f^{(0)}, A_H^{(0)}, \beta^{(0)}, f^{(1)}, A_H^{(1)}, \beta^{(1)}, f^{(2)}, A_H^{(2)}, \beta^{(2)} \ldots, f^{(N)}, A_H^{(N)}, \beta^{(N)})$

Sample Ripple Frequency In a standard Gibbs sampler, the samples $f^{(k)}$, $A_H^{(k)}$ and $\beta^{(k)}$ are drawn directly from the distribution $\Pi(f|\mathbf{X}, A_H^{(k-1)}, \beta^{(k-1)})$, $\Pi(A_H|\mathbf{X}, f^{(k)}, \beta^{(k-1)})$, and $\Pi(\beta|\mathbf{X}, f^{(k)}, A_H^{(k)})$, respectively. As the three distributions for $f^{(k)}$, $A_H^{(k)}$, and $\beta^{(k)}$ can-

Algorithm 5 *A Metropolis-Hastings sampler that generates* $f^{(k)} \sim \Pi(f|\mathbf{X}, A_H^{(k-1)}, \beta^{(k-1)})$

1: Generate a candidate f^c from $p(f^c|f^{old} = f^{(k-1)})$ (Equation (A.25))
2: Generate a random number u from $\mathscr{U}(0,1)$
3: **if** $u \leq a(f^c, f^{old} = f^{(k-1)})$ (Equation (A.26)) **then**
4: Accept f^c, $f^{(k)} = f^c$
5: **else**
6: Reject f^c, $f^{(k)} = f^{(k-1)}$
7: **end if**

not be drawn from directly, we use a single Metropolis-Hastings step to sample these values. As in a standard Metropolis-Hastings algorithm, a proposal distribution is used to generate sample candidates. Then, a candidate can be accepted as the next sample or rejected according to its acceptance ratio. The Metropolis-Hastings step is summarized in Algorithm 5. In the current implementation, the proposal distribution used to generate a frequency candidate, f^c, for the sample $f^{(k)}$ is a Gaussian mixture centered on the previous frequency sample value, $f^{(k-1)}$,

$$p(f^c|f^{(k-1)}) = \quad w_{n,f}\mathcal{N}(f^c|f^{(k-1)}, \quad s_{n,f}) + w_{w,f}\mathcal{N}(f^c|f^{(k-1)}, s_{w,f}) \qquad \text{(A.25)}$$

where $w_{n,f}$ and $w_{w,f}$ are fixed parameters such that $w_{w,f} + w_{n,f} = 1$ and $w_{n,f}, w_{w,f} \geq 0$. These are used to determine the relative frequency sampling from a Gaussian whose variance is either relatively small (narrow spread: $s_{n,f}$) or large (wide spread: $s_{w,f}$). The variance values, $s_{n,f}$ and $s_{w,f}$, are fixed values used to generate the frequency samples. When sampling from the narrow Gaussian mixture component, a small area in the parameter space surrounding the current frequency sample is explored to refine the current frequency estimate. Sampling from the proposal distribution using the wide Gaussian mixture component allows for larger exploration of the parameter space. Therefore, the $w_{n,f}, w_{w,f}, s_{n,f}$ and $s_{w,f}$ values will affect the speed of convergence of the proposed method by balancing

the local versus global search parameters. Given this proposal distribution, the acceptance ratio for the frequency values used to evaluate f^c will be

$$\begin{aligned} a &= \frac{\Pi(f^c|\mathbf{X}, A_H^{(k-1)}, \beta^{(k-1)})}{p(f^c|f^{(k-1)})} \frac{p(f^{(k-1)}|f^c)}{\Pi(f^{(k-1)}|\mathbf{X}, A_H^{(k-1)}, \beta^{(k-1)})} \\ &= \frac{\Pi(f^c|\mathbf{X}, A_H^{(k-1)}, \beta^{(k-1)})}{\Pi(f^{(k-1)}|\mathbf{X}, A_H^{(k-1)}, \beta^{(k-1)})} \end{aligned} \tag{A.26}$$

where

$$\Pi(f|\mathbf{X}, A_H, \beta) \propto \prod_{i=1}^{N} \mathcal{N}\left(x_{i,r}|a^*(r, A_H, f, \beta), \sigma_x^2\right) \mathcal{N}_t(f|\mu_f, \sigma_f, l_f, u_f) \tag{A.27}$$

with $N_t(f|\mu_f, \sigma_f, l_f, u_f)$ as the truncated Gaussian prior distribution in (A.21). The second equality is the result of the proposal distribution being a symmetric distribution. For the first or only pass over an area, $N_t(f|\mu_f, \sigma_f, l_f, u_f)$ will be replaced by the uniform prior distribution in (A.18).

Sample Ripple Amplitude The sampling step for the ripple field amplitude parallels the step for sampling the frequency using the latest sampled frequency value. Again, the proposal distribution is a Gaussian mixture centered on the previous A_H value,

$$p(A_H^c|A_H^{(k-1)}) = w_{n,A}\mathcal{N}(A_H^c|A_H^{(k-1)}, s_{n,A}) + w_{w,A}\mathcal{N}(A_H^c|A_H^{(k-1)}, s_{w,A}) \tag{A.28}$$

where $w_{n,A}$ and $w_{w,A}$ are fixed parameters such that $w_{w,A} + w_{n,A} = 1$ and $w_{n,A}, w_{w,A} \geq 0$. Given that the proposal distribution is symmetric, the acceptance ratio used is

$$a = \frac{\Pi(A_H^c|\mathbf{X}, f^{(k)}, \beta^{(k-1)})}{\Pi(A_H^{(k-1)}|\mathbf{X}, f^{(k)}, \beta^{(k-1)})} \tag{A.29}$$

where

$$\Pi(A_H|\mathbf{X}, f, \beta) \propto \prod_{i=1}^{N} \mathscr{N}\left(x_{i,r}|a^*(r, A_H, f, \beta), \sigma_x^2\right) \cdot \mathscr{N}_t(A_H|\mu_{A_H}, \sigma_{A_H}, l_{A_H}, u_{A_H}). \tag{A.30}$$

Sample Ripple Orientation In accordance with the frequency and amplitude sampling, the proposal distribution is a Gaussian mixture centered on the previous β value,

$$p(\beta^c|\beta^{(k-1)}) = w_{n,\beta}\mathscr{N}(\beta^c|\beta^{(k-1)}, s_{n,\beta}) + w_{w,\beta}\mathscr{N}(\beta^c|\beta^{(k-1)}, s_{w,\beta}) \tag{A.31}$$

where $w_{n,\beta}$ and $w_{w,\beta}$ are fixed parameters such that $w_{w,\beta} + w_{n,\beta} = 1$ and $w_{n,\beta}, w_{w,\beta} \geq 0$. Given that the proposal distribution is symmetric, the acceptance ratio used is

$$a = \frac{\Pi(\beta^c|\mathbf{X}, f^{(k)}, A_H^{(k)})}{\Pi(\beta^{(k-1)}|\mathbf{X}, f^{(k)}, A_H^{(k)})} \tag{A.32}$$

where

$$\Pi(\beta|\mathbf{X}, f, A_H) \propto \prod_{i=1}^{N} \mathscr{N}\left(x_{i,r}|a^*(r, A_H, f, \beta), \sigma_x^2\right) \cdot \mathscr{N}_t(\beta|\mu_\beta, \sigma_\beta, l_\beta, u_\beta). \tag{A.33}$$

Parallel Sampling

The proposed method needs a large number of samples to estimate the parameters of interest. This can be time-consuming when dealing with a large number of images. Therefore, we consider parallelizing the proposed method by allowing separate processors to generate samples of f, A_H or β thus shortening the execution time to generate some fixed number of samples. In our implementation, each processor is assigned to generate the same number of samples. The parameter estimates returned by our implementation are those that were generated with the largest posterior value. Since we are only returning the single sample with the largest posterior, several sampling chains may be run in parallel and a single sample selected among the chains. However, this approach does not decrease any "burn-in" period of the sampler. Also, if the mean, variance or any other statistic of the samples need to be estimated, this approach cannot be used. Instead, parallel MCMC based on methods such as regeneration should be utilized [69, 70].

Initialization and Parameter Settings

The proposed sampling method requires several parameters to be set prior to running the algorithm. For our current implementation and all of the experimental results shown, initialization for the algorithm and parameters are determined using the following methods.

Initialization Prior to sampling, the SAS image being analyzed is smoothed and normalized. To preserve the edges which usually define the boundary between nonshadowed and shadowed region in SAS imagery, a 2-D median filter is employed to remove the noise without blurring edges. Smoothing is done by applying a 2-D median filter to the image

and is done to minimize the speckle term in (A.1) (since the pixel scattering strength, $a(r)$, is the term of interest). After smoothing, the image is normalized by subtracting the mean image pixel value and dividing by the maximum image pixel value.

Initial values for the frequency are determined by first computing the frequency of each row using 1D-DFT and, then, using the averaged frequency as the initial estimate. The SAS image being considered contains the input pixels $\mathbf{X}$. The A_H values are initialized to the root mean square of the pixel intensity values in the SAS image. The initial values of β are estimated using the Hough line transform method [71]. As shown in Figure A.4, the boundaries between shadowed and unshadowed region can be interpreted as lines, whose angles can be considered as approximations to ripple orientation. From the Hough transform method, we use the average of the detected line angles as the initial value for β.

Parameter Settings In the current implementation, the phase value, b, is fixed to zero. When sampling the ripple frequency, amplitude, and orientation, the parameters of their corresponding truncated prior Gaussian distributions need to be set. The upper and lower truncation points are set to constrain the ripple parameters to physically possible values. In our current implementation, these lower and upper truncation points are set to [0.6, 1.5] cycle/m for f, [0, 0.06] m for A_H, and [-π, π] for β. However, these values can be easily modified given further constraining prior information (such as prior constrained information derived from oceanographic data).

Parameters for the proposal distributions for f, A_H, and β include $w_{n,f}$, $w_{n,A}$, $w_{n,\beta}$, $s_{n,f}$, $s_{n,A}$, $s_{n,\beta}$, $s_{w,f}$, $s_{w,A}$, and $s_{w,\beta}$. In the current implementation, these values are set to 0.7, 0.7, 0.7, 0.01, 0.01, 0.01, 0.1, 0.1, and 0.1, respectively. Given that the proposed method is a Metropolis-within-Gibbs algorithm, it will benefit from the corresponding con-

vergence guarantees. However, the speed of convergence is dependent on these parameter settings. Future work can include investigating methods to optimize these proposal distribution parameters.

When given only an initial single pass of the data, uninformative priors are used for both of these parameters of interest (as opposed to the truncated Gaussian distributions defined in the previous section). These uninformative priors can be approximated by setting the σ_f, σ_{A_H}, and σ_β value to extremely large variance values relative to the truncation points of the prior. However, when previous aspects and passes over an area of interest have been analyzed, the ripple frequency, amplitude, and orientation estimates obtained from the previous passes can be used to set the prior distribution parameter settings. In these cases, μ_f, μ_{A_H}, and μ_β, can be set to the values estimated from previous passes and, σ_f, σ_{A_H}, and σ_β can be set based on the confidence of these previous estimates.

A.4 Experimental Results

The proposed sampling method was applied to simulated and real SAS imagery. Results are shown on these data sets and discussed in the following sections.

A.4.1 Image Size Analysis

The first set of experiments examines the role the size of an input image plays. An input image should be large enough to provide enough pixels with statistical consistency to extract the desired sand ripple parameters. However, an image must also be small enough to avoid too much spatial variability across the image sand ripple field since, in this work, (A_H, f, β) are assumed to be constant within an image clip. This experiment aims to determine the

lower bound on the image size needed to estimate sand ripple parameters.

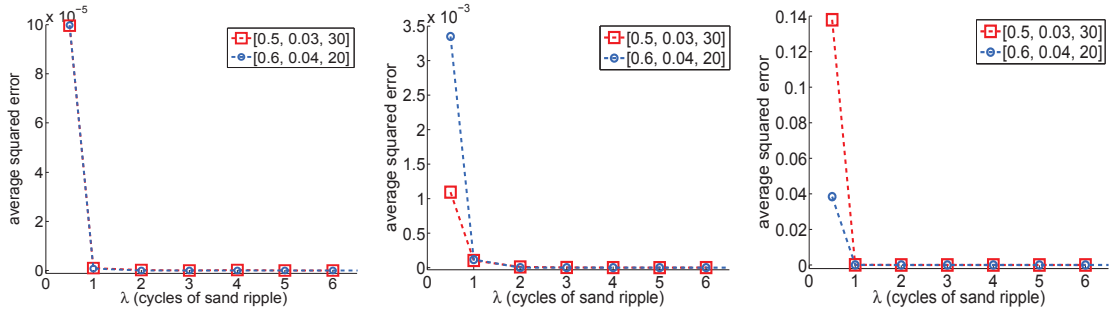
The proposed method was run 10 times on two sets of simulated data generated using the model defined in (A.1) with the expanded pixel scattering strength model shown in (A.3), and the Gaussian model as the scattering cross section. The speckle term, $Z(x, y)$ in (A.1) was generated using a Rayleigh distributed speckle with a parameter value of $s = 1 \times 10^2$. One set of the simulated data is generated using the parameter setting $(A_H, f, \beta) = (0.03 \text{ m}, 0.5 \text{ cycle/m}, \pi/6)$, and another one $(A_H, f, \beta) = (0.04 \text{ m}, 0.6 \text{ cycle/m}, \pi/9)$. Each set contains seven square images with a different number of sand ripple cycles, $\lambda = \{0.5, 1, 2, 3, 4, 5, 6\}$. The width of each image is determined as $\lfloor \frac{\lambda}{f \cdot R} \rfloor$, where R is the sonar range resolution.

Figure A.6. (a) - (c) show the average squared error (over 10 runs) of the estimated A_H, f, and β values versus number of sand ripple cycles, respectively. The average squared error stays stable after the "elbow points" in these sub-figures. It seems reasonable to conclude that the statistics of the estimates stay consistent when the image covers the area with no less than one ripple. Since the minimum ripple frequency f used in our work is 0.5 cycle/m and the sonar range resolution R is assumed to be 0.025m, the image width is required to be greater than 160 pixels so that at least two ripples are present in the image, thus the statistical consistency is guaranteed.

A.4.2 Simulated SAS Data - One Pass

In this experiment, three sets of simulated data were generated based on the Gaussian scattering cross section, the sin^2, and the SSA models. The proposed method was run using sequential and parallel Metropolis-within-Gibbs samplers.

For comparison, a traditional frequency estimation approach where imagery were trans-



(a) A_H (b) f (c) β

Figure A.6: Average squared error of the estimated (a) ripple amplitude A_H, (b) ripple frequency f, and (c) ripple orientation β under varying number of cycles of sand ripple. The curve with red square markers is for the simulated image with $(f, A_H, \beta) = (0.5 \text{ cycle/m}, 0.03 \text{ m}, \pi/6)$, and the curve with blue circle markers is for the simulated image with $(f, A_H, \beta) = (0.6 \text{ cycle/m}, 0.04 \text{ m}, \pi/9)$

formed from the spatial domain into the frequency domain [72, 73] was also employed. A 1D-DFT method is used, which computes the frequency of each image row and takes the average over all row frequencies as the estimate ripple frequency. In our experiments, the sonar range resolution R is assumed to be 0.025m, and the simulated images are sized to be 400×400. The DFT length N_{DFT} is 400, and the DFT frequency resolution is $1/(N_{DFT} \cdot R) = 0.1$ cycle/m. Note that in the 1D-DFT approach, only frequency can be estimated. The proposed method is able to estimate ripple frequency, amplitude and orientation (using the mode of the resulting estimated distributions, i.e., the samples with the largest likelihood value). The proposed method was also compared to a geometry-based method of amplitude estimation (as illustrated in Figure A.7). In the geometry-based method, the shadowed regions are first segmented through thresholding, where a pixel is identified as shadowed if its intensity value is lower than a predefined threshold value. The complete shadowed regions are selected to compute the amplitude (the first and last shadowed region are excluded). The estimated amplitude for each complete shadowed region is

defined as

$$\hat{A}_H \simeq \frac{\Delta r_0}{2r_0} A_S, \tag{A.34}$$

where Δr_0 is the width of the selected shadowed region, and r_0 is the distance from the shadowed region's end point to the origin. The average of the estimated amplitudes are used as the final estimated amplitude.

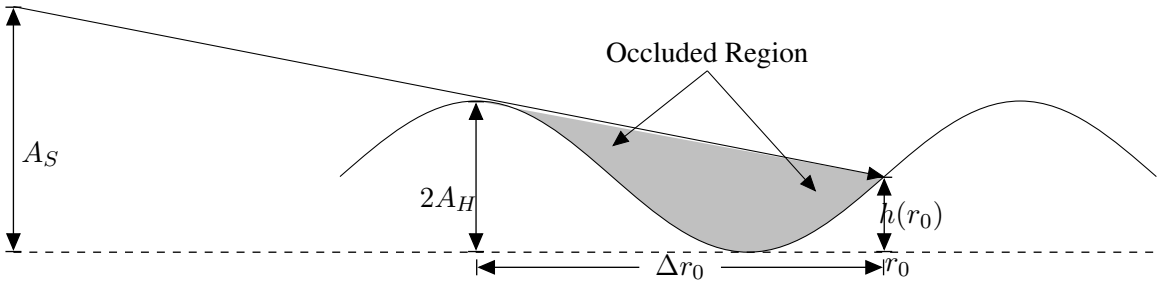


Figure A.7: The geometry-based method of amplitude estimation. $h(r_0)$ is approximated to be zero.

To compare the proposed method with the DFT approach and the geometry-based method and to evaluate the performance of the parallelized sampling method, we have conducted a comprehensive series of experiments, obtaining a large amount of experimental results. Several selected representative experimental results that compare the proposed method to the other methods and examine the performance of parallelized sampling method are shown here. As can be seen in these results, the proposed approach outperforms the traditional DFT approach and the geometry-based method. Furthermore, the parallel sampling approach provides results with a similar error rate but with a significantly shorter running time.

The complete set of experiments are described in the following sub-sections.

Varying the true f value

In this experiment, the f values were varied to be 0.5, 0.6, 0.7 and 0.8 cycle/m. As the DFT frequency resolution is 0.1 cycle/m, these four ripple frequencies are distinguishable in the 1D-DFT method. The A_H value was set to 0.03 m, the β value is $\frac{\pi}{6}$, and the image clip varied from 0.2 m to 10.2 m in range. In all simulated experiments, the height of the SAS array was fixed to 4.0 m. For each f value, the algorithm was run 10 times and a new simulated image was generated for each run.

Varying the true A_H value

The A_H values were varied to be 0.02, 0.03, 0.04 and 0.05m while the f value was set to 0.7 cycle/m, the β value is $\frac{\pi}{6}$, and the image clip varied from 0.2m to 10.2m in range. Again, for each A_H value, the algorithm was run 10 times and a new simulated image was generated for each run. The experimental results using the Gaussian model with a sequential sampling method (as shown in Table A.1) wass selected to show the comparison between the proposed method and the DFT and geometry-based methods. The average squared error of our approach (sequential) is much lower than that of the 1D-DFT method in sand ripple frequency estimation, and much lower than that of the geometry-based method in sand ripple amplitude estimation.

Varying the true β value

The β values were varied to be $\frac{\pi}{6}$, $\frac{\pi}{4}$, $\frac{\pi}{3}$, and $\frac{5}{12}\pi$ while the f value was set to 0.7 cycle/m, the A_H value is 0.03 m, and the image clip varied from 0.2 m to 10.2 m in range. Again, for each β value, the algorithm was run 10 times and a new simulated image was generated for each run. The experimental results using the sin^2 model (as shown in Table A.2) is

Table A.1: One Pass Simulated Data-Gaussian Model: Varying A_H, Average Squared Error($\pm$ Standard Deviation)

True A_H		0.02 m	0.03 m	0.04 m	0.05 m
Seq.	A_H	3.9×10^{-9}	3.0×10^{-8}	2.7×10^{-8}	4.3×10^{-8}
		(3.8×10^{-9})	(3.8×10^{-8})	(2.4×10^{-8})	(6.4×10^{-8})
	f	2.2×10^{-8}	5.3×10^{-9}	3.0×10^{-9}	1.3×10^{-8}
		(6.0×10^{-8})	(4.8×10^{-9})	(3.0×10^{-9})	(2.6×10^{-8})
	β	6.6×10^{-10}	5.8×10^{-10}	7.2×10^{-10}	3.2×10^{-10}
		(4.8×10^{-10})	(9.3×10^{-10})	(1.2×10^{-9})	(4.2×10^{-10})
DFT	f	3.7×10^{-2}	3.6×10^{-2}	3.6×10^{-2}	3.6×10^{-2}
		(1.9×10^{-1})	(1.9×10^{-1})	(1.9×10^{-1})	(1.9×10^{-1})
Geo.	A_H	1.6×10^{-1}	1.5×10^{-1}	1.5×10^{-1}	1.3×10^{-1}
		(0.0)	(0.0)	(2.9×10^{-17})	(0.0)

The units of the average squared error is m^2 for A_H, $(cycle/m)^2$ for f, and rad^2 for β.

selected to evaluate the performance of the parallel version of our approach. The parallel sampling approach shortens the execution time by approximately a factor of 4 (the number of processors) and still achieves similar performance to the sequential sampling in terms of average squared error ($\pm$ standard deviation).

Varying the range of the simulated SAS imagery

The range values were varied to be beginning from 7.5 m in range to 30.0 m in range while the A_H value was set to 0.03 m and the f value was fixed to 0.7 cycle/m. For each range, the algorithm was run 10 times and a new simulated image was generated for each run. The amplitude estimation results of the Gaussian model, the sin^2 model, and the SSA model using the sequential sampling method are shown in Table A.3. The proposed method has similar performances on all three models, illustrating that the proposed method can be applied to any appropriate given model.

Table A.2: One Pass Simulated Data-sin^2 Model: Varying β, Average Squared Error($\pm$ Standard Deviation)

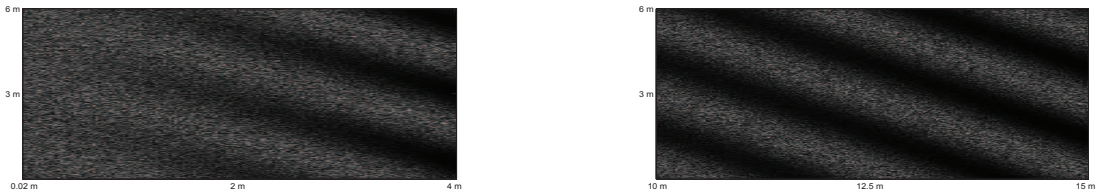
True β		$\frac{\pi}{6}$	$\frac{\pi}{4}$	$\frac{\pi}{3}$	$\frac{5}{12}\pi$
Seq.	A_H	1.8×10^{-8}	1.3×10^{-7}	3.6×10^{-8}	1.2×10^{-6}
		(2.3×10^{-8})	(3.2×10^{-7})	(5.0×10^{-8})	(3.5×10^{-6})
	f	8.6×10^{-9}	1.5×10^{-8}	1.7×10^{-7}	3.0×10^{-8}
		(9.3×10^{-9})	(2.6×10^{-8})	(3.0×10^{-7})	(2.8×10^{-8})
	β	1.8×10^{-9}	8.6×10^{-9}	1.5×10^{-7}	6.0×10^{-8}
		(4.0×10^{-9})	(1.5×10^{-8})	(2.3×10^{-7})	(5.5×10^{-8})
	Exec.	615.4 seconds	615.0 seconds	606.6 seconds	608.2 seconds
Par.	A_H	2.3×10^{-8}	3.5×10^{-7}	1.0×10^{-7}	2.2×10^{-7}
		(3.4×10^{-8})	(2.9×10^{-7})	(1.3×10^{-7})	(4.6×10^{-7})
	f	2.9×10^{-8}	3.1×10^{-7}	1.8×10^{-7}	5.7×10^{-7}
		(4.6×10^{-8})	(4.6×10^{-7})	(1.6×10^{-7})	(8.4×10^{-7})
	β	5.6×10^{-9}	3.8×10^{-8}	1.9×10^{-7}	1.1×10^{-6}
		(6.3×10^{-9})	(5.4×10^{-8})	(7.2×10^{-8})	(1.4×10^{-6})
	Exec.	153.4 seconds	152.8 seconds	152.4 seconds	152.1 seconds

Table A.3: One Pass Simulated Data-Three Models: Varying Range, Average Squared Error($\pm$ Standard Deviation) of A_H

Beginning Range	7.5 m	15.0 m	22.5 m	30.0 m
Gau	3.5×10^{-10}	6.3×10^{-9}	6.0×10^{-9}	8.0×10^{-9}
	(7.9×10^{-10})	(3.9×10^{-9})	(4.4×10^{-9})	(2.1×10^{-9})
sin^2	3.2×10^{-10}	2.3×10^{-9}	4.2×10^{-9}	4.6×10^{-9}
	(9.8×10^{-10})	(3.0×10^{-9})	(3.5×10^{-9})	(3.5×10^{-9})
SSA	2.4×10^{-9}	4.0×10^{-9}	2.2×10^{-9}	3.0×10^{-9}
	(5.1×10^{-9})	(3.1×10^{-9})	(3.3×10^{-9})	(3.1×10^{-9})

A.4.3 Ripple Orientation Estimation using Hough Line Transform Method

As mentioned in Section A.3.2, the Hough line transform method was used for β initialization. For experiments in Section A.4.2 where obvious occluded regions were present in the simulated imagery, the Hough line transform method could extract the occluded region boundaries precisely, thus providing an accurate initialization. The initial value of β was more accurate than the final estimated β in these cases. However, the performance of Hough transform approach depends on the occurrence of occluded regions in SAS imagery. In this section, the proposed method was compared with the Hough transform method on images with insignificant occluded regions (as shown in Figure A.8.(a)). The β values were varied to be $\frac{\pi}{6}$, $\frac{\pi}{4}$, $\frac{\pi}{3}$, and $\frac{5}{12}\pi$ while the f value was set to 0.5 cycle/m, the A_H value is 0.03 m. The image clip varied from 0.2 m to 4.0 m with insignificant occluded regions. For each β value, the algorithm was run 10 times and a new simulated image was generated for each run. Results are shown in Table A.4. The Hough line transform method achieved higher average squared error than the proposed approach when the occluded regions were insignificant.



(a) Insignificant occluded region (b) Significant occluded region

Figure A.8: Image clips with insignificant and significant occluded regions.

Table A.4: Varying β, Average Squared Error($\pm$ Standard Deviation)

True β		$\frac{\pi}{6}$	$\frac{\pi}{4}$	$\frac{\pi}{3}$	$\frac{5}{12}\pi$
Seq.	β	8.1×10^{-8} (1.2×10^{-8})	2.2×10^{-8} (5.0×10^{-8})	3.8×10^{-8} (4.3×10^{-8})	5.7×10^{-8} (1.0×10^{-7})
Hough	β	5.4×10^{-4} (0.0)	3.0×10^{-4} (0.0)	1.4×10^{-4} (0.0)	3.4×10^{-5} (0.0)

A.4.4 Simulated SAS Data-Multiple Pass

In the first experiment, we have no prior knowledge of the ripple parameters other than knowing physically reasonable values. So, for those experiments we use a uniform prior distribution such that each possible value is treated equally. However, for multiple passes, we can impose prior knowledge from the estimated result of the first pass. By establishing an informative prior distribution for the subsequent passes, the estimated result can be further refined. In this experiment, the informative prior distribution we use is a truncated Gaussian prior distribution centered at the estimated result of the preceding pass (Equation (A.20)-(A.22)). The σ_f, σ_{A_H},and σ_β values were set to be 3.00×10^{-6}, based on the squared error computed from simulated data experiments with corresponding parameter settings.

Consider the following two-pass experiment. Simulated data for the same area is generated from two different ranges (7.5 m-27.5 m, 15.0 m-35.0 m). First, the proposed sampling algorithm with the uniform prior distribution (as described in Equations (A.17)-(A.19)) is applied to the first pass of the data with a collection range of 7.5 m-27.5 m. Then, the second pass (15.0 m-35.0 m) of the area is analyzed using both the uniform prior distribution (uninformative) and the truncated Gaussian distribution whose parameters are set based on the estimated result of the first pass (informative). In our experiments, we choose four sets of values for (A_H, f, β), $(0.02, 0.8, \pi/4)$, $(0.03, 0.8, \pi/4)$, $(0.02, 0.9, \pi/6)$ and

Table A.5: Two-pass Simulated Data: Average Squared Error ($\pm$ Standard Deviation)

True (A_H, f, β)		(0.02, 0.8, $\pi/4$)	(0.03, 0.8, $\pi/4$)	(0.02, 0.9, $\pi/6$)	(0.03, 0.9, $\pi/6$)
A_H	Pass 1	2.7×10^{-8}	9.5×10^{-9}	2.9×10^{-8}	3.3×10^{-8}
		(4.4×10^{-8})	(1.3×10^{-8})	(4.6×10^{-8})	(2.4×10^{-8})
	Pass	5.9×10^{-8}	1.5×10^{-8}	1.6×10^{-8}	3.3×10^{-8}
	2-Uninf.	(4.8×10^{-8})	(2.2×10^{-8})	(2.1×10^{-8})	(3.1×10^{-8})
	Pass	$\mathbf{1.2 \times 10^{-8}}$	$\mathbf{9.4 \times 10^{-9}}$	$\mathbf{1.0 \times 10^{-8}}$	$\mathbf{1.5 \times 10^{-8}}$
	2-Inf.	(2.2×10^{-8})	(1.2×10^{-8})	(1.1×10^{-8})	(2.3×10^{-8})
f	Pass 1	$\mathbf{1.4 \times 10^{-8}}$	$\mathbf{3.2 \times 10^{-8}}$	4.5×10^{-8}	1.3×10^{-7}
		(1.9×10^{-8})	(5.0×10^{-8})	(5.9×10^{-8})	(2.1×10^{-7})
	Pass	6.3×10^{-8}	9.4×10^{-8}	8.4×10^{-8}	1.3×10^{-6}
	2-Uninf.	(7.1×10^{-8})	(1.1×10^{-7})	(1.8×10^{-7})	(3.1×10^{-6})
	Pass	1.5×10^{-8}	3.6×10^{-8}	$\mathbf{4.2 \times 10^{-8}}$	$\mathbf{1.3 \times 10^{-7}}$
	2-Inf.	(2.1×10^{-8})	(6.7×10^{-8})	(5.9×10^{-8})	(2.2×10^{-7})
β	Pass 1	5.0×10^{-9}	6.3×10^{-9}	4.9×10^{-9}	2.0×10^{-9}
		(6.8×10^{-9})	(8.6×10^{-9})	(7.1×10^{-9})	(4.2×10^{-9})
	Pass	3.0×10^{-8}	2.3×10^{-8}	8.7×10^{-8}	5.3×10^{-8}
	2-Uninf.	(3.0×10^{-8})	(4.1×10^{-8})	(2.5×10^{-7})	(1.7×10^{-7})
	Pass	$\mathbf{4.5 \times 10^{-9}}$	$\mathbf{5.9 \times 10^{-9}}$	$\mathbf{4.5 \times 10^{-9}}$	$\mathbf{1.9 \times 10^{-9}}$
	2-Inf.	(6.6×10^{-9})	(7.0×10^{-9})	(5.9×10^{-9})	(3.5×10^{-9})

$(0.03, 0.9, \pi/6)$. For each set, the experiment was run 10 times and the results are shown in Table A.5. As seen in Table A.5, the results of the second pass tend to improve overall. This is because more information is available from the second pass due to more regions of occlusion. Imagery that contains more regions of occlusion provides more information when estimating ripple parameters. As the second pass was at larger range values, there were larger regions of occlusion. Furthermore, for two different passes over the same area, two different occlusion regions from the same ripple peaks can be observed. This provides more collaborating information about the height of the ripple field which is leveraged by the sampling algorithm to improve the overall estimate. Future work will investigate setting prior distribution values based on the relative rates of occlusion in the input imagery.

A.4.5 Measured SAS Data

The proposed approach was also applied to real, measured SAS imagery. Both single pass and multiple pass experiments were conducted. In the case of real SAS imagery, ground truth information in terms of true ripple frequency, amplitude, and orientation are unknown. Thus, results are shown with comparison to simulated data generated using the estimated parameter values.

One Pass

For the one-pass experiments, five real image clips were considered. Three image clips have zero orientation, ranging from 17.1 m to 31.7 m. The other two clips have non-zero orientation, ranging from 20.0 m to 33.2 m. These experiments were run using the Gaussian scattering cross section model. In Figure A.9 and A.10, we show the image clips used, the resulting estimated parameters, and simulated imagery generated using the resulting ripple frequency and amplitude.

By qualitatively comparing the simulated imagery (without any speckle) to the measured data, we can see that the best results are obtained when the sand ripples do not merge or split. This is due to fixing the phase value, b to be zero. Future work will incorporate phase estimates for every pixel in the image (which will result in a considerable increase in the size of the search space). Despite not estimating b, the estimated values for the real imagery appear qualitatively good.

Multiple Passes

For the multiple pass experiment, two SAS images collected over the same area from different ranges (21.9 m-36.6 m and 40.9 m-50.0 m) were used to estimate the ripple

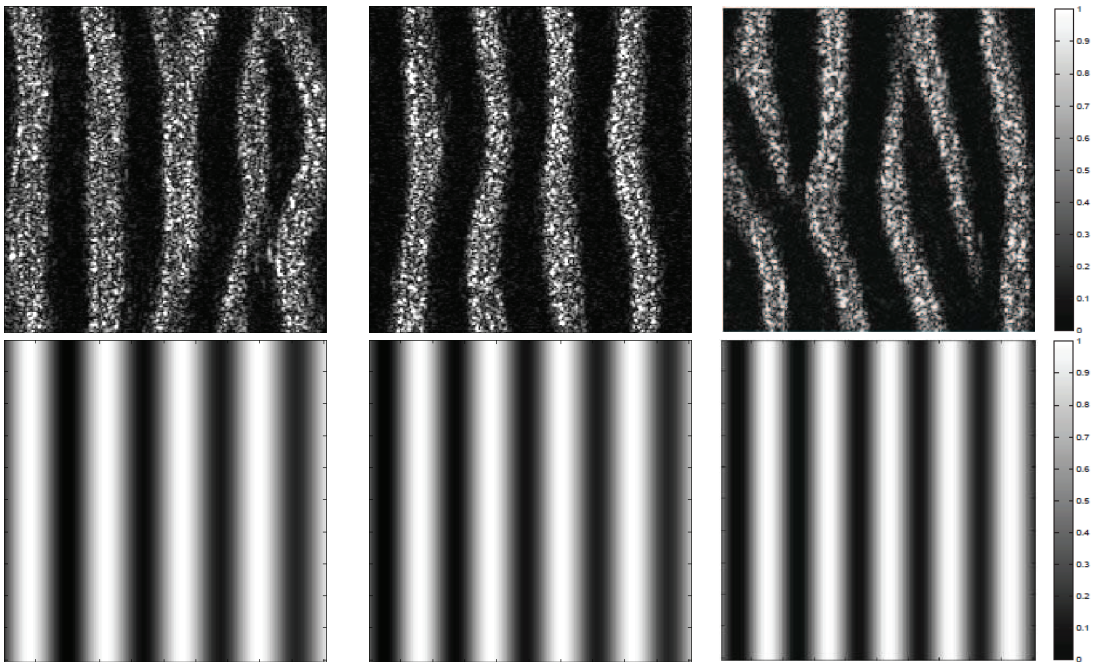


Figure A.9: (a) - (c) are three real image clips. Their corresponding simulated imagery (without speckle) generated using the estimated frequency, and amplitude parameters are shown in (d) - (f). The estimated results: (a) Estimated $f = 1.05$ cycle/m, $A_H = 0.035$ m, (b) Estimated $f = 0.93$ cycle/m, $A_H = 0.017$ m, (c) Estimated $f = 1.05$ cycle/m, $A_H = 0.016$ m.

frequency and amplitude. Both of these images are centered over the same sand ripple area with slightly different window sizes. These images were collected several days apart. These figures are shown in Figure A.11. The first pass initially estimated values of $f = 1.464 \pm 0.0092$, $A_H = 0.0595 \pm 0.0004$ (average value over ten runs $\pm$ standard deviation). These values were then used as prior values while incorporating the second pass to refine the estimates. After the second pass, the estimated parameter values for the region were $f = 1.465 \pm 0.0093$, $A_H = 0.0598 \pm 0.0002$ (average value over ten runs $\pm$ standard deviation).

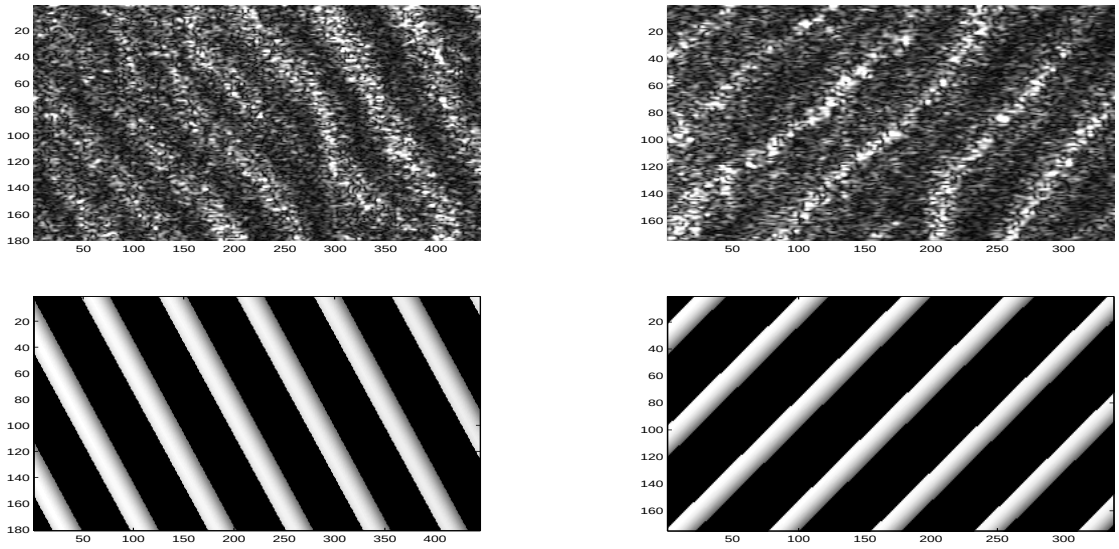


Figure A.10: (a) and (b) are two real image clips. Their corresponding simulated imagery (without speckle) generated using the estimated frequency, amplitude, and orientation parameters are shown in (c) and (d). The estimated results: (a) Estimated $f = 1.14$ cycle/m, $A_H = 0.017$ m, $\beta = 24.9°$, (b) Estimated $f = 1.22$ cycle/m, $A_H = 0.019$ m, $\beta = -32.6°$.

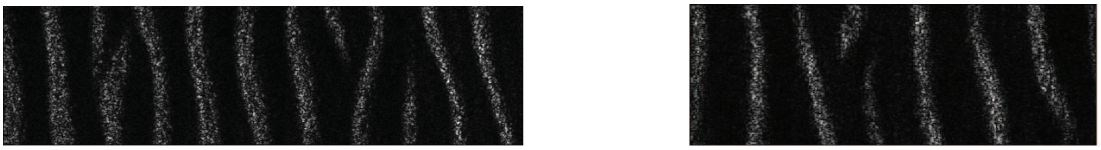


Figure A.11: First (a) and second (b) pass of rippled sand region collected at 21.9 m to 36.6 m and 40.9 m to 50.0 m in range, respectively.

A.5 Summary and Future Work

This work extends Lyons' sand ripple model by incorporating occluded/shadowed regions due to the preceding peaks in a ripple field and presents a hierarchical Bayesian framework and MCMC sampling method for sand ripple field frequency, amplitude, and orientation estimation from SAS imagery. Furthermore, the proposed sampling method is parallelized to improve running time. Experimental results show the effectiveness of the extended Lyons'

model. When compared with a 1D-DFT method for frequency estimation, a geometry-based method for amplitude estimation, and the Hough transform method for orientation estimation, the proposed sampling method is shown to outperform the others. In comparison, the three sand ripple parameters are simultaneously estimated and have much higher estimation accuracy. The parallel version of the proposed sampling method significantly shortens the execution time by approximately a factor of the number of processors used while preserving the estimation accuracy. Furthermore, multiple pass experiments show a performance gain by using informative prior information from previous passes over an area.

Future work will include many significant extensions to the proposed approach. The current implementation assumes a flat (non-sloping) seafloor. Future work will include expanding the assumed models and sampling approach to incorporate sloping sea-floors. This extension will add additional invariant parameters for estimation (namely, the slope of the sea-floor). Furthermore, sand ripple fields do not have a constant phase value but, instead, ripples merge and split. Therefore, future work will also include estimation of the phase value, b, for every SAS pixel under consideration. Since the extended sand ripple model is based on Lyons' model which is only applicable to high frequency and narrow angle SAS systems, a more general model suitable to other SAS systems will also be investigated.

Appendix B

Appendix 2 - Invariant Parameter Estimation Across Varying Seabeds in Synthetic Aperture Sonar Imagery

In this work, we seek to estimate invariant features of the seafloor to describe and distinguish between seafloor types. Our approach makes the assumption that the local characteristics of the true bathymetry of the seafloor can be represented by a Gaussian Markov Random Field (GMRF). Since we characterize the bathymetry of the seafloor, this method yields GMRF parameters that are invariant to acoustic sensing modality and geometry.

The intensity-based features can be degraded due to the speckle noise present in the sonar imagery and vary drastically with the sensing geometry. Given a particular sensing geometry and bathymetry, the textures and pixel values imaged by a SAS system will vary over the same spatial region of the seafloor. Features based on image characteristics are often not quantifying invariant parameters of the seafloor but, instead, are describing the relationship of that particular sensing geometry with respect to that area of the seafloor.

A different pass of the SAS sensor with varying range and aspect angle over the same region may result in very different image-based features. In this work, we investigate the use of Gaussian Markov Random Fields (GMRFs) to characterize the underlying bathymetry depicted in a SAS image. In order to estimate the GMRF parameters that describe the bathymetry of a region, bathymetry slope values for imaged seafloor are needed. However, many of the pixels on the seafloor are shadowed and, thus, much of the information related to their heights relative to neighboring pixels is unknown. To address this issue, an alternating optimization (AO) algorithm is used to estimate the GMRF parameters. The proposed method can be divided into four main steps summarized in Figure B.1.

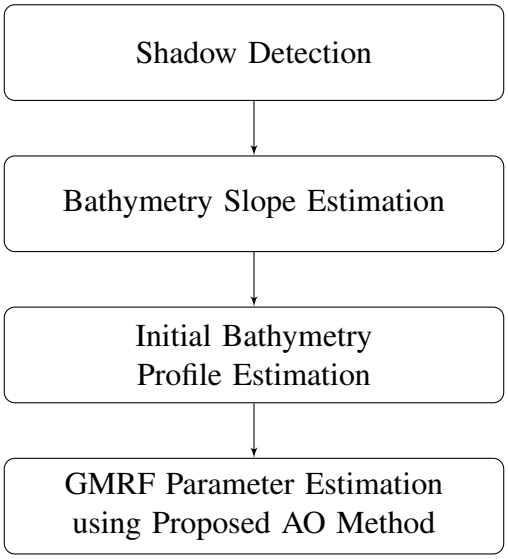


Figure B.1: Flowchart of the proposed GMRF method.

B.1 Estimating the Relative Bathymetry Profile

Incident acoustic waves of SAS will be reflected and scattered from the seafloor. The component that is backscattered towards the sonar will be collected by the sonar receivers and

contribute to the formation of the SAS imagery. Here, SAS imagery have been modeled with an intensity representation of the backscattered acoustic energy given a specific range, depression and aspect angle [49]. Given knowledge of the backscattering model with appropriate assumptions, the intensity image can be converted into the bathymetry map[74]. A challenge in this conversion comes from the shadowed pixels where the incident acoustic wave is blocked by the preceding contour and, thus, no bathymetry information are available for these pixels. In our work, we determine locations of the shadowed pixels based on the local mean and variance and compute an estimate of slope of the bathymetry for non-shadowed pixels using the variation of local pixel intensity.

B.1.1 Shadow Detection

Our current approach for identifying occluded pixels is to apply a local mean and variance filter to the input image. Shadowed/occluded regions have both a small local mean and low variance. Thus, after computing the mean and variance in a local window $\mathscr{W}$ surrounding each pixel, the mean and variance values are thresholded with predetermined, fixed values. If both the mean and variance are below their respective thresholds, the pixel is labeled as a shadowed or occluded pixel.

B.1.2 Backscattering Model: Lambertian Reflectance Model

In order to estimate the bathymetry, we need a backscattering model relating intensity to bathymetry. Seafloor backscattering is a complicated phenomenon, which is generally considered to be composed of a combination of surface and volume scattering, (i.e., roughness interface scattering and scattering from inhomogeneities within the sediment volume

[75, 76]). However, for simplicity, in our work a Lambertian reflectance model is applied as the backscattering model [76–78]. Under the Lambertian reflectance model, image intensity $I_{i,j}$ is the cosine of the angle between the seafloor orientation (normal vector) $\mathbf{N}_{i,j}$ and the incident acoustic wave direction $\mathbf{L}_{i,j}$ at the seafloor location corresponding to pixel (i, j)[78]. This model is illustrated in Figure B.2.

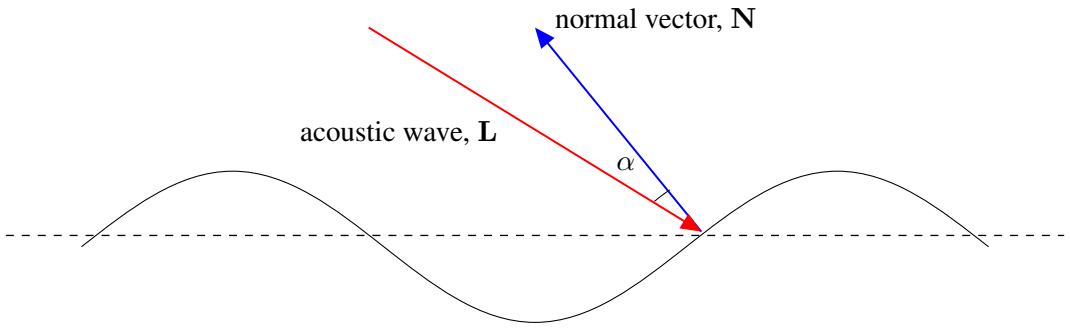


Figure B.2: Graphic depiction of the Lambertian reflectance model.

The SAS image intensity can be expressed as

$$I_{i,j} = \cos(\alpha) = \frac{\langle \mathbf{L}_{i,j}, \mathbf{N}_{i,j} \rangle}{\langle \mathbf{L}_{i,j}, \mathbf{L}_{i,j} \rangle \cdot \langle \mathbf{N}_{i,j}, \mathbf{N}_{i,j} \rangle}, \tag{B.1}$$

with

$$\mathbf{L}_{i,j} = \begin{pmatrix} j \\ 0 \\ -A_H + H_{i,j} \end{pmatrix} \simeq \begin{pmatrix} j \\ 0 \\ -A_H \end{pmatrix}, \text{ and } \mathbf{N}_{i,j} = \begin{pmatrix} -\partial H/\partial j \\ -\partial H/\partial i \\ 1 \end{pmatrix} \simeq \begin{pmatrix} -\Delta H_{i,j}/\Delta R \\ 0 \\ 1 \end{pmatrix}, \tag{B.2}$$

where A_H is the sonar height, $H_{i,j}$ is the absolute bathymetry value corresponding to pixel (i, j), ΔR is the sonar range resolution and $\Delta H_{i,j}$ is the bathymetry slope at (i, j). In our data, A_H is usually around $5m$ and $H_{i,j}$ is generally smaller than $10cm$ ($H_{i,j} \ll A_H$). Thus,

the component $-A_H + H_{i,j}$ in $\mathbf{L}_{i,j}$ can be approximated by $-A_H$. Using this approximation yields an expression that depends only on relative change in $H_{i,j}$ rather than absolute bathymetry values. As described in the following Section B.1.3, $\Delta H_{i,j}$ can be estimated using (B.1)-(B.2) given $I_{i,j}$, A_H and ΔR.

B.1.3 Initial Bathymetry Estimation

Since input SAS imagery are generally not calibrated, intensity information does not provide any absolute bathymetry information. Therefore, only bathymetry slope information can be estimated from the input image. Pixel intensity is an indicator of slope information of the seafloor, which can potentially provide a relative change in height at each pixel [79]. A large intensity value generally indicates a steeper seafloor slope (e.g., maximal values are perpendicular to the grazing angle from the sensor). A small value indicates a shallow seafloor slope. Given an assumed value for the first pixel in each row of the image and the first pixel following any occluded region, a bathymetry slope profile for all of the observed pixels can be estimated based on the relative intensity of each pixel. Essentially, all intensity values for observed pixels are normalized and used as a slope estimate (change from the bathymetry value of the preceding pixel) to estimate the bathymetry profile of observed pixels. In this initial estimate, shadowed pixels are set to have a bathymetry slope of -2. From (B.1) - (B.2) and this naive interpolation scheme, the bathymetry slope can be estimated as

$$\Delta H_{i,j} = \begin{cases} \dfrac{\Delta R}{\tan(\arctan(\frac{A_H}{j}) + \arccos(I_{i,j}))} & \text{if } M_{i,j} = 0, \\ -2 & \text{otherwise,} \end{cases} \tag{B.3}$$

where $M_{i,j}$ is the shadow map at pixel (i, j) such that $M_{i,j} = 1$ if pixel (i, j) is shadowed/occluded and $M_{i,j} = 0$ otherwise. Given (B.3), the estimated bathymetry profile will be

$$\hat{H}_{i,j} = \sum_{k=1}^{j} \Delta H_{i,k}, \tag{B.4}$$

which is the cumulative sum of the bathymetry slope along the range direction. The first column in the image is assumed to be the reference height, which means that $\Delta H_{i,1} = I_{i,1}$.

B.2 GMRF Parameter Estimation and Bathymetry Slope Refinement

Once an initial bathymetry profile for the observed pixels is estimated as described above, then the GMRF parameters for that bathymetry profile can be estimated. We use an alternating optimization (AO) with an Iterated Conditional Modes (ICM)[80, 81] approach. In our approach, we assume that our initial estimated bathymetry profile is inaccurate. Thus, the proposed algorithm aims both to refine the bathymetry profile and update the GMRF parameters. The algorithm is summarized in the pseudo-code in Algorithm 6. Each step of the proposed approach is described in the following sub-sections as indicated in Algorithm 6.

B.2.1 Bathymetry Profile Distortion Model Parameter Estimation

The bathymetry profile estimate is refined by considering this as an image restoration problem using the current estimated GMRF model [82, 83]. In our work, due to occurrence of shadowed pixels, we consider two types of possible bathymetry profile errors, namely,

Algorithm 6 *Proposed AO with ICM algorithm*

Input: Shadow map $M_{i,j}$ and initial bathymetry map $\hat{H}$.

1: Initialize σ_0^2, σ_1^2, the bathymetry map, H, and the GMRF parameters, $\boldsymbol{\beta}$.
2: Estimate σ_1^2 by $\hat{\sigma}_1^2 = \arg\max_{\sigma_1^2} P(\hat{H}|H, \sigma_1^2)$ (Section B.2.1).
3: Estimate σ_0^2 by $\hat{\sigma}_0^2 = \arg\max_{\sigma_0^2} P(\hat{H}|H, \sigma_0^2)$ (Section B.2.1).
4: Estimate $\boldsymbol{\beta}$ by $\bar{\boldsymbol{\beta}} = \arg\max_{\boldsymbol{\beta}} P(H|\boldsymbol{\beta})$. MPLE is used to maximize the PL likelihood function (Section B.2.2).
5: Update H by $H = \arg\max_H P(H|\hat{H}, \boldsymbol{\beta}, \sigma_0^2, \sigma_1^2)$ based on the current $\hat{H}$, $\boldsymbol{\beta}$, σ_0^2 and σ_1^2. The maximization here is performed by using ICM method (Section B.2.3).
6: Return to step 2 for a fixed number of iterations or until convergence.

occlusion error and reconstruction error. Occlusion error is found at the shadowed pixels where any bathymetry information has been concealed and, thus, bathymetry is initially only estimated by the naive interpolation scheme described in Equation (B.3). Reconstruction error represents all the other possible errors such as noise in the sonar image or error due to the backscattering model. We treat the occlusion error and reconstruction error as two zero-mean Gaussian random variables with different variances. Given this model, the bathymetry profile distortion model can be expressed as

$$\hat{H}_{i,j} = H_{i,j} + e_{i,j}, \tag{B.5}$$

where $e_{i,j} \sim \mathcal{N}(0, \sigma_{i,j}^2)$ is the independent zero-mean Gaussian error which distorts the true bathymetry map, H. We assume that occlusion error occurs at only shadowed pixels and reconstruction error occurs only at the remaining pixels,

$$e_{i,j} \sim \begin{cases} \mathcal{N}(0, \sigma_1^2) & \text{if } M_{i,j} = 1, \\ \mathcal{N}(0, \sigma_0^2) & \text{otherwise,} \end{cases} \tag{B.6}$$

where $e_{i,j} \sim \mathcal{N}(0, \sigma_1^2)$ represents occlusion error and $e_{i,j} \sim \mathcal{N}(0, \sigma_0^2)$ reconstruction error.

Following the conditional independence assumption [80] that given H, all the elements in $\hat{H}$ are conditionally independent and that each element of $\hat{H}_{i,j}$ has a known conditional density function $\mathcal{N}(\hat{H}_{i,j}|H_{i,j}, \sigma_{i,j}^2)$, then the conditional density of $\hat{H}$ given H can be simplified to

$$\begin{aligned} P(\hat{H}|H) &= \prod_{\forall(i,j)} p(\hat{H}_{i,j}|H_{i,j}, M_{i,j}) = \prod_{\substack{(i,j):\\ M_{i,j}=1}} p(\hat{H}_{i,j}|H_{i,j}) \prod_{\substack{(i,j):\\ M_{i,j}=0}} p(\hat{H}_{i,j}|H_{i,j}) \\ &= \prod_{\substack{(i,j):\\ M_{i,j}=1}} \mathcal{N}(\hat{H}_{i,j}|H_{i,j}, \sigma_1^2) \prod_{\substack{(i,j):\\ M_{i,j}=0}} \mathcal{N}(\hat{H}_{i,j}|H_{i,j}, \sigma_0^2). \end{aligned} \tag{B.7}$$

Then, the distortion model parameters σ_1^2 and σ_0^2 can be estimated using Maximum likelihood estimation (MLE). By maximizing (B.7), the update equations for σ_1^2 and σ_0^2 are

$$\hat{\sigma}_1^2 = \frac{1}{m_1} \sum_{\substack{(i,j):\\ M_{i,j}=1}} [H_{i,j} - \hat{H}_{i,j}]^2, \tag{B.8}$$

and

$$\hat{\sigma}_0^2 = \frac{1}{m_0} \sum_{\substack{(i,j):\\ M_{i,j}=0}} [H_{i,j} - \hat{H}_{i,j}]^2, \tag{B.9}$$

respectively, where m_1 is the number of shadowed pixels and m_0 non-shadowed pixels.

B.2.2 Gaussian Markov Random Fields (GMRFs) Model Parameter Estimation

In step 4 of Algorithm 1, the GMRF parameters, $\boldsymbol{\beta}$, are updated. As stated previously, these parameters are the desired invariant seafloor features. In our implementation, we adopt a 2nd-order homogeneous GMRF model with four spatial interaction parameters $\boldsymbol{\beta} = (\beta_1, \beta_2, \beta_3, \beta_4)^T$. As shown in Figure B.3, β_1 and β_2 are for neighbors which are one pixel apart horizontally and vertically, and β_3 and β_4 for diagonally adjacent neighbors in southeast and southwest directions [84]. The local conditional probability density function (pdf) is defined as

$$H_{i,j}|H_{\mathcal{N}_{i,j}} \sim \mathcal{N}\left((w_{i,j}H)^T\boldsymbol{\beta}, \sigma^2\right), \tag{B.10}$$

with

$$(w_{i,j}H) = [(H_{i,j-1} + H_{i,j+1}), (H_{i-1,j} + H_{i+1,j}), (H_{i-1,j-1} + H_{i+1,j+1}), (H_{i+1,j-1} + H_{i-1,j+1})]^T, \tag{B.11}$$

where $w_{i,j}$ is an indicator matrix which selects the neighbors of pixel (i, j) and σ^2 is the local variance which is usually assumed to be known.

For GMRF parameter estimation, Maximum likelihood estimation (MLE) is computationally demanding particularly for large lattices. Besag [85, 86] proposed the Maximum pseudo-likelihood estimation (MPLE) as an alternative to MLE, where the pseudo-likelihood (PL) is defined as the product of the local conditional pdfs of each pixel given the neighbors [84–87]. In our approach, we employ the MPLE method to estimate the $\boldsymbol{\beta}$ GMRF parameters. Note that the MPLE approach is used to estimate the $\boldsymbol{\beta}$ parameters

only. The log-PL is defined as

$$\mathbf{ln}PL(H) = -\frac{1}{2}MNln(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{N}\sum_{j=1}^{M}(H_{i,j} - (w_{i,j}H)^T\boldsymbol{\beta})^2, \tag{B.12}$$

and, given this model, the GMRF parameter update equations are

$$\bar{\boldsymbol{\beta}} = \left(\sum_{i=1}^{N}\sum_{j=1}^{M}(w_{i,j}H)(w_{i,j}H)^T\right)^{-1}\left(\sum_{i=1}^{N}\sum_{j=1}^{M}(w_{i,j}H)H_{i,j}\right), \tag{B.13}$$

and

$$\bar{\sigma^2} = \frac{1}{MN}\sum_{i=1}^{N}\sum_{j=1}^{M}\left(H_{i,j} - (w_{i,j}H)\bar{\boldsymbol{\beta}}\right)^2. \tag{B.14}$$

β_3	β_2	β_4
β_1		β_1
β_4	β_2	β_3

Figure B.3: The 2nd-order Neighborhood Structure

B.2.3 Bathymetry Profile Refinement

After parameter estimation for the distortion and GMRF models, the proposed approach refines the bathymetry profile according to all available information; that is, find the H that maximizes the conditional probability $p(H|\hat{H})$. Probabilistic solutions for this problem, such as simulated annealing, the Gibbs sampler, or the Metropolis algorithm, ensure the convergence toward a global maximum but have computational costs [88]. Thus, we employ a deterministic algorithm called Iterated Conditional Modes (ICM) [80] which lightens

the computational burden by providing a suboptimal solution that sequentially maximizes local conditional probabilities, $p\left(H_{i,j}|\hat{H}, H_{\mathcal{N}_{i,j}}\right)$.

From the assumptions that the observed $\hat{H}_{i,j}$ are conditionally independent given H, and that each $\hat{H}_{i,j}$ has the conditional density function $p\left(\hat{H}_{i,j}|H_{i,j}\right)$ dependent only on $H_{i,j}$, it follows that

$$p\left(H_{i,j}|\hat{H}, H_{\mathcal{N}_{i,j}}\right) \propto p\left(\hat{H}_{i,j}|H_{i,j}\right) p\left(H_{i,j}|H_{\mathcal{N}_{i,j}}\right). \tag{B.15}$$

Maximizing Equation (B.15) is equivalent to minimizing the the following potential

$$V_{i,j} = \frac{1}{2\sigma_{i,j}^2}\left(H_{i,j} - \hat{H}_{i,j}\right)^2 + \frac{1}{2\sigma^2}\left(H_{i,j} - \sum_{i,k\in\mathcal{N}_{i,j}} \beta_{i,k} H_{i,k}\right)^2, \tag{B.16}$$

where the first term enforces the assumption that our distortion model is Gaussian distributed and the second term minimizes the difference between neighboring pixels. ICM evaluates the second term given $H_{i,k}$ from the last iteration. Using the method of Lagrange multipliers, the update equation for $H_{i,j}$ at iteration $(t+1)$ is given by

$$H_{i,j}^{(t+1)} = \frac{\sigma^2}{\sigma_{i,j}^2 + \sigma^2}\hat{H}_{i,j} + \frac{\sigma_{i,j}^2}{\sigma_{i,j}^2 + \sigma^2} \sum_{i,k\in\mathcal{N}_{i,j}} \beta_{i,k} H_{i,k}^{(t)}. \tag{B.17}$$

B.3 Experiments

In order to evaluate our estimated GMRF parameter, two sets of experiments were conducted. The first experiment examined the ability for the proposed algorithm to recover true bathymetry GMRF parameters given varying sonar aspect angles. The second experiment

examined the ability of the proposed method to estimate parameters that can distinguish between different sea floor types.

B.3.1 Experiment I - Invariant to SAS System Height

The first set of experiments investigates the stability of the approach to estimate the correct GMRF parameter while varying SAS system aspect angles. With a fixed underlying true bathymetry map, SAS imagery were simulated. Then, the proposed AO approach was applied to each simulated image and the estimated GMRF parameters were compared to the true values.

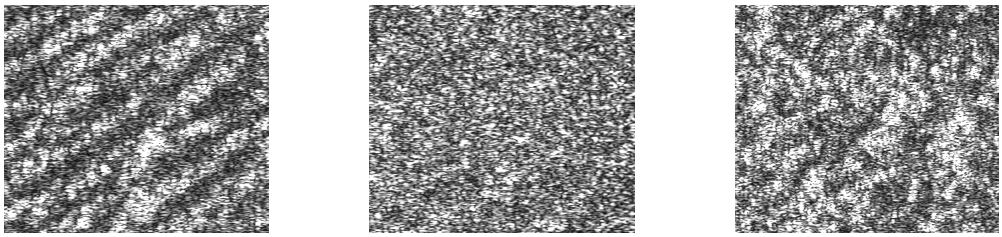
In this experiment, the sonar system was set to three different heights resulting three different sensing geometries. Each sensing geometry was used to simulated SAS imagery with two different GMRF parameter sets, namely, [0.5108, 0.4107, 0.2097, 0.2097], and [0.5432, 0.2910, 0.1580, 0.1580]. For each parameter set, 10 simulations of 513-by-513 images were generated. Then, given these simulated data set, the proposed algorithm was applied and the GMRF parameters were estimated. In Table B.1 and Table B.2, the average squared error (standard deviation) is computed over 10 runs. Our proposed algorithm (denoted by AO(s)) is compared to three other methods, namely, MPLE, Coding method and AO(e). This experiment shows that, even with varying sensing geometry, stable GMRF parameters for the underlying bathymetry are estimated. With respect to the estimation error, our proposed AO algorithm achieves the best performance among the four methods.

B.3.2 Experiment II

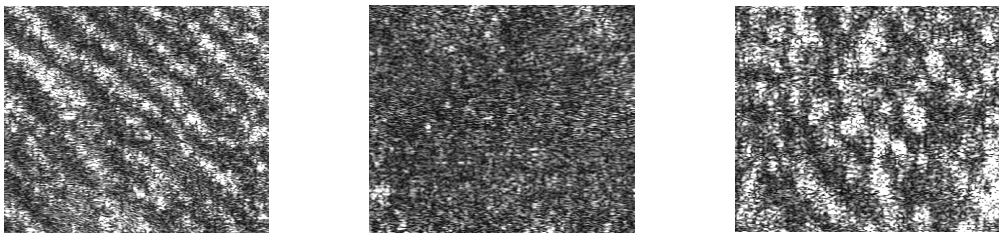
The second set of experiments investigates the value of the GMRF parameters with respect to distinguishing between different seabed types. In this experiment, GMRF parameters are estimated from several manually selected and labeled sub-images from real SAS data. Since our data set was unlabeled with respect to seafloor type, the following labeling process was used. Each extracted sub-image from real SAS data was labeled by three individuals separately. The true label was determined to be the seafloor type that was agreed by all the three individuals. Samples in which all three individuals did not assign the same label were removed from the data set. Samples of two seafloor types considered are shown in Figure B.4. The dataset contains 104 sub-images categorized into two classes: sand ripple and hard-pack sand. The sand ripple class has 60 sub-images and the hard-pack sand class has 44 sub-images. For each sub-image, the GMRF parameters were estimated with the proposed approach. These GMRF parameters were then used as features in KNN classifier trained to distinguish between the two seafloor types. For the KNN classifier, we varied K from 3 to 9 with step-size 2 and the decision rule was majority-vote. The performance was measured using a two-fold cross-validation scheme. The sub-images are assigned into two equal size sets. Each set is consist of the randomly selected half samples of the two classes. 1000 rounds of validations are performed, and the validation results are averaged over the rounds.

A practical problem in this experiment is downsampling of the sub-images. SAS images have very high resolution (e.g., on the order of centimeters per pixel). The physical distance between adjacent pixels in SAS imagery is very small, which indicates a high order GMRF with a large amount of parameters. To reduce the computational complexity and make the images fit with the proposed 2nd-order GMRF, the sub-images were downsampled.

Since the optimal downsampling rate is hard to determine, we conducted experiments in which the downsampling rate was varied from 1 to 10. Experimental results are shown in Table B.3. The best classification results occur when the downsampling rate is 5 and the classification accuracy at this rate is around 80%. This experiment validates the value of the GMRF parameters to describe varying seafloor types.



(a) sand ripple 1 (b) hard-pack sand 1 (c) seagrass 1



(d) sand ripple 2 (e) hard-pack sand 2 (f) seagrass 2

Figure B.4: Samples of different seabed types.

B.4 Summary and Future Work

This work presents a set of invariant and discriminative features describing the local characteristics of the bathymetry map and proposes an AO with ICM method to extract these features. The proposed AO with ICM algorithm provides reliable GMRF parameters, and the developed feature is shown to be invariant to the sensing geometry and applicable to seafloor classification tasks.

Our future work will include many extensions to and further investigation of the proposed method. For example, the current distortion is modeled to be Gaussian distributed. Future work will include expanding the distortion model to consider more applicable error distributions. ICM methods depends highly on initialization and investigation into a better initialization scheme will be developed in the future. Furthermore, investigation can be conducted into the applicability of the GMRF model on the range of possible seabed bathymetry profiles as well as the applicability of the proposed invariant features extracted from Interferometric SAS.

Table B.1: Average Squared Error (Standard Deviation) of Estimates Over 10 Simulations of the 1st Parameter Set.

Meth.	SP	$\beta_1 = 0.5108$	$\beta_2 = 0.4107$
AO(s)	P_1	$\mathbf{1.06 \times 10^{-4}}(2.93 \times 10^{-5})$	$\mathbf{1.96 \times 10^{-4}}(2.50 \times 10^{-3})$
	P_2	$\mathbf{1.05 \times 10^{-4}}(2.30 \times 10^{-5})$	$1.65 \times 10^{-4}(1.70 \times 10^{-3})$
	P_3	$\mathbf{1.07 \times 10^{-4}}(1.79 \times 10^{-5})$	$\mathbf{2.67 \times 10^{-6}}(1.40 \times 10^{-3})$
AO(e)	P_1	$1.07 \times 10^{-4}(2.89 \times 10^{-5})$	$1.20 \times 10^{-3}(2.20 \times 10^{-3})$
	P_2	$1.06 \times 10^{-4}(2.23 \times 10^{-5})$	$\mathbf{1.48 \times 10^{-4}}(1.60 \times 10^{-3})$
	P_3	$1.08 \times 10^{-4}(1.48 \times 10^{-5})$	$4.07 \times 10^{-4}(1.20 \times 10^{-3})$
MPLE	P_1	$1.14 \times 10^{-4}(1.49 \times 10^{-5})$	$3.04 \times 10^{-2}(1.60 \times 10^{-3})$
	P_2	$1.14 \times 10^{-4}(6.95 \times 10^{-6})$	$2.19 \times 10^{-2}(1.40 \times 10^{-3})$
	P_3	$1.15 \times 10^{-4}(6.38 \times 10^{-6})$	$1.80 \times 10^{-2}(1.20 \times 10^{-3})$
Coding	P_1	$1.14 \times 10^{-4}(1.51 \times 10^{-5})$	$3.04 \times 10^{-2}(1.60 \times 10^{-3})$
	P_2	$1.14 \times 10^{-4}(7.03 \times 10^{-6})$	$2.19 \times 10^{-2}(1.40 \times 10^{-3})$
	P_3	$1.15 \times 10^{-4}(6.48 \times 10^{-6})$	$1.80 \times 10^{-2}(1.20 \times 10^{-3})$
Meth.	SP	$\beta_3 = 0.2097$	$\beta_4 = 0.2097$
AO(s)	P_1	$\mathbf{1.25 \times 10^{-4}}(1.30 \times 10^{-3})$	$\mathbf{1.20 \times 10^{-4}}(1.50 \times 10^{-3})$
	P_2	$\mathbf{6.79 \times 10^{-6}}(6.58 \times 10^{-4})$	$\mathbf{6.09 \times 10^{-4}}(1.40 \times 10^{-3})$
	P_3	$\mathbf{1.20 \times 10^{-5}}(8.28 \times 10^{-4})$	$\mathbf{1.72 \times 10^{-5}}(1.00 \times 10^{-3})$
AO(e)	P_1	$4.70 \times 10^{-4}(1.10 \times 10^{-3})$	$4.57 \times 10^{-4}(1.30 \times 10^{-3})$
	P_2	$9.91 \times 10^{-5}(6.16 \times 10^{-4})$	$1.09 \times 10^{-4}(1.30 \times 10^{-3})$
	P_3	$1.92 \times 10^{-4}(8.68 \times 10^{-4})$	$2.13 \times 10^{-4}(9.13 \times 10^{-4})$
MPLE	P_1	$8.40 \times 10^{-3}(1.10 \times 10^{-3})$	$8.30 \times 10^{-3}(1.00 \times 10^{-3})$
	P_2	$6.10 \times 10^{-3}(7.15 \times 10^{-4})$	$6.20 \times 10^{-3}(1.20 \times 10^{-3})$
	P_3	$5.00 \times 10^{-3}(7.84 \times 10^{-4})$	$5.10 \times 10^{-3}(9.28 \times 10^{-4})$
Coding	P_1	$8.40 \times 10^{-3}(1.10 \times 10^{-3})$	$8.30 \times 10^{-3}(1.00 \times 10^{-3})$
	P_2	$6.10 \times 10^{-3}(7.15 \times 10^{-4})$	$6.20 \times 10^{-3}(1.20 \times 10^{-3})$
	P_3	$5.00 \times 10^{-3}(7.84 \times 10^{-4})$	$5.10 \times 10^{-3}(9.31 \times 10^{-4})$

AO(s) is the proposed method. AO(e) is similar to AO(s) but treats reconstruction and occlusion error equally, i.e., $\sigma_0^2 = \sigma_1^2$. MPLE is maximum pseudo-likelihood estimation method [85] and Coding is the Coding method [89].

Table B.2: Average Squared Error (Standard Deviation) of Estimates Over 10 Simulations of the 2nd Parameter Set.

Meth.	SP	$\beta_1 = 0.5432$	$\beta_2 = 0.2910$
AO(s)	P_1	$\mathbf{1.80 \times 10^{-3}}(2.95 \times 10^{-5})$	$\mathbf{1.51 \times 10^{-5}}(3.60 \times 10^{-3})$
	P_2	$\mathbf{1.80 \times 10^{-3}}(4.22 \times 10^{-5})$	$1.10 \times 10^{-3}(3.40 \times 10^{-3})$
	P_3	$\mathbf{1.80 \times 10^{-3}}(1.52 \times 10^{-5})$	$5.15 \times 10^{-4}(2.90 \times 10^{-3})$
AO(e)	P_1	$1.80 \times 10^{-3}(2.91 \times 10^{-5})$	$3.42 \times 10^{-4}(3.20 \times 10^{-3})$
	P_2	$1.80 \times 10^{-3}(4.07 \times 10^{-5})$	$\mathbf{7.27 \times 10^{-5}}(3.10 \times 10^{-3})$
	P_3	$1.80 \times 10^{-3}(1.51 \times 10^{-5})$	$\mathbf{1.13 \times 10^{-5}}(3.30 \times 10^{-3})$
MPLE	P_1	$1.90 \times 10^{-3}(1.43 \times 10^{-5})$	$1.76 \times 10^{-2}(2.50 \times 10^{-3})$
	P_2	$1.90 \times 10^{-3}(1.89 \times 10^{-5})$	$1.23 \times 10^{-2}(2.40 \times 10^{-3})$
	P_3	$1.90 \times 10^{-3}(8.51 \times 10^{-6})$	$1.01 \times 10^{-2}(2.60 \times 10^{-3})$
Coding	P_1	$1.80 \times 10^{-3}(1.42 \times 10^{-5})$	$1.76 \times 10^{-2}(2.50 \times 10^{-3})$
	P_2	$1.90 \times 10^{-3}(1.88 \times 10^{-5})$	$1.23 \times 10^{-2}(2.40 \times 10^{-3})$
	P_3	$1.90 \times 10^{-3}(8.33 \times 10^{-6})$	$1.01 \times 10^{-2}(2.60 \times 10^{-3})$
Meth.	SP	$\beta_3 = 0.1580$	$\beta_4 = 0.1580$
AO(s)	P_1	$\mathbf{1.47 \times 10^{-4}}(2.30 \times 10^{-3})$	$\mathbf{1.29 \times 10^{-4}}(1.80 \times 10^{-3})$
	P_2	$\mathbf{1.85 \times 10^{-5}}(1.60 \times 10^{-3})$	$\mathbf{2.26 \times 10^{-5}}(2.20 \times 10^{-3})$
	P_3	$\mathbf{3.99 \times 10^{-6}}(1.70 \times 10^{-3})$	$\mathbf{4.26 \times 10^{-6}}(1.80 \times 10^{-3})$
AO(e)	P_1	$4.75 \times 10^{-4}(2.20 \times 10^{-3})$	$4.47 \times 10^{-4}(1.40 \times 10^{-3})$
	P_2	$7.22 \times 10^{-5}(1.60 \times 10^{-3})$	$7.11 \times 10^{-5}(2.00 \times 10^{-3})$
	P_3	$1.49 \times 10^{-4}(2.00 \times 10^{-3})$	$1.50 \times 10^{-4}(1.80 \times 10^{-3})$
MPLE	P_1	$6.30 \times 10^{-3}(1.80 \times 10^{-3})$	$6.10 \times 10^{-3}(1.20 \times 10^{-3})$
	P_2	$4.60 \times 10^{-3}(1.40 \times 10^{-3})$	$4.60 \times 10^{-3}(1.60 \times 10^{-3})$
	P_3	$3.90 \times 10^{-3}(1.70 \times 10^{-3})$	$3.90 \times 10^{-3}(1.50 \times 10^{-3})$
Coding	P_1	$6.30 \times 10^{-3}(1.80 \times 10^{-3})$	$6.10 \times 10^{-3}(1.20 \times 10^{-3})$
	P_2	$4.60 \times 10^{-3}(1.40 \times 10^{-3})$	$4.60 \times 10^{-3}(1.70 \times 10^{-3})$
	P_3	$3.90 \times 10^{-3}(1.70 \times 10^{-3})$	$3.90 \times 10^{-3}(1.40 \times 10^{-3})$

Table B.3: Validation Results with Different K and Downsampling Rates

Downsampling Rate	$1(\%)$	$2(\%)$	$3(\%)$	$4(\%)$	$5(\%)$
$K = 3$	$56.12(55.55)$	$54.06(54.23)$	$62.04(61.62)$	$76.87(75.29)$	$\mathbf{79.38(77.82)}$
$K = 5$	$57.20(56.20)$	$56.06(54.89)$	$62.77(61.57)$	$79.09(77.49)$	$\mathbf{80.45(78.71)}$
$K = 7$	$58.25(57.09)$	$55.70(54.73)$	$62.15(60.43)$	$79.87(78.39)$	$\mathbf{80.97(79.12)}$
$K = 9$	$58.05(56.82)$	$55.56(54.14)$	$61.23(59.90)$	$81.04(79.15)$	$\mathbf{81.21(79.71)}$
Downsampling Rate	$6(\%)$	$7(\%)$	$8(\%)$	$9(\%)$	$10(\%)$
$K = 3$	$76.53(75.42)$	$78.68(77.74)$	$78.60(77.80)$	$76.32(75.09)$	$76.18(74.59)$
$K = 5$	$78.15(76.94)$	$79.67(78.04)$	$79.41(77.85)$	$74.05(72.41)$	$72.22(70.87)$
$K = 7$	$78.91(77.42)$	$80.01(78.62)$	$79.21(77.86)$	$71.79(70.10)$	$68.41(67.16)$
$K = 9$	$79.16(77.74)$	$80.92(78.65)$	$79.32(77.59)$	$70.61(69.58)$	$65.72(64.56)$

Numbers outside parentheses are classification accuracy for the first fold, and numbers in parentheses are for the second fold.

Bibliography

[1] D. M. Blei, "Probabilistic topic models," *Communications of the ACM*, vol. 55, no. 4, pp. 77–84, 2012.

[2] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.

[3] C. Reed, "Latent dirichlet allocation: Towards a deeper understanding," 2012.

[4] M. Steyvers and T. Griffiths, "Probabilistic topic models," *Handbook of Latent Semantic Analysis*, vol. 427, no. 7, pp. 424–440, 2007.

[5] T. L. Griffiths and M. Steyvers, "Finding scientific topics," *Proceedings of the National Academy of Sciences*, vol. 101, no. suppl 1, pp. 5228–5235, 2004.

[6] J. Huang and T. Malisiewicz, "Correlated topic model details," Carnegie Mellon University, 2006, Tech. Rep.

[7] D. M. Blei and J. D. Lafferty, "Dynamic topic models," in *Proceedings of the International Conference on Machine Learning*, 2006, pp. 113–120.

[8] J. Chang and D. M. Blei, "Relational topic models for document networks," in *International Conference on Artificial Intelligence and Statistics*, 2009, pp. 81–88.

[9] C. Wang, B. Thiesson, C. Meek, and D. M. Blei, "Markov topic models," in *International Conference on Artificial Intelligence and Statistics*, 2009, pp. 583–590.

[10] X. Wei, J. Sun, and X. Wang, "Dynamic mixture models for multiple time-series." in *International Joint Conference on Artificial Intelligence*, vol. 7, 2007, pp. 2909–2914.

[11] C. Wang, D. Blei, and D. Heckerman, "Continuous time dynamic topic models," *arXiv preprint arXiv:1206.3298*, 2012.

[12] X. Wang and A. McCallum, "Topics over time: a non-markov continuous-time model of topical trends," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2006, pp. 424–433.

[13] J. D. Mcauliffe and D. M. Blei, "Supervised topic models," in *Advances in Neural Information Processing Systems*, 2008, pp. 121–128.

[14] J. Zhu, A. Ahmed, and E. P. Xing, "Medlda: maximum margin supervised topic models for regression and classification," in *Proceedings of the International Conference on Machine Learning*, 2009, pp. 1257–1264.

[15] D. Ramage, D. Hall, R. Nallapati, and C. D. Manning, "Labeled lda: A supervised topic model for credit attribution in multi-labeled corpora," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2009, pp. 248–256.

[16] A. Acharya, A. Rawal, R. J. Mooney, and E. R. Hruschka, "Using both latent and supervised shared topics for multitask learning," in *Machine Learning and Knowledge Discovery in Databases*. Springer, 2013, pp. 369–384.

[17] T. Hofmann, "Probabilistic latent semantic indexing," in *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1999, pp. 50–57.

[18] J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman, and W. T. Freeman, "Discovering object categories in image collections," 2005.

[19] 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 *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, 2006, pp. 1605–1614.

[20] L. Fei-Fei and P. Perona, "A bayesian hierarchical model for learning natural scene categories," in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, 2005, pp. 524–531.

[21] P. Quelhas, F. Monay, J.-M. Odobez, D. Gatica-Perez, T. Tuytelaars, and L. Van Gool, "Modeling scenes with local descriptors and latent aspects," in *IEEE International Conference on Computer Vision*, vol. 1, 2005, pp. 883–890.

[22] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman, "Learning object categories from google's image search," in *IEEE International Conference on Computer Vision*, vol. 2, 2005, pp. 1816–1823.

[23] J. C. van Gemert, C. J. Veenman, A. W. Smeulders, and J.-M. Geusebroek, "Visual word ambiguity," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 7, pp. 1271–1283, 2010.

[24] D. Weinshall, G. Levi, and D. Hanukaev, "Lda topic model with soft assignment of descriptors to words," in *Proceedings of the International Conference on Machine Learning*, 2013, pp. 711–719.

[25] E. B. Sudderth, A. Torralba, W. T. Freeman, and A. S. Willsky, "Learning hierarchical models of scenes, objects, and parts," in *IEEE International Conference on Computer Vision*, vol. 2, 2005, pp. 1331–1338.

[26] X. Wang and E. Grimson, "Spatial latent dirichlet allocation," in *Advances in Neural Information Processing Systems*, 2008, pp. 1577–1584.

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

[28] B. Zhao, L. Fei-Fei, and E. P. Xing, "Image segmentation with topic random field," in *European Conference on Computer Vision*. Springer, 2010, pp. 785–798.

[29] K. A. Heller, S. Williamson, and Z. Ghahramani, "Statistical models for partial membership," in *Proceedings of the International Conference on Machine Learning*, 2008, pp. 392–399.

[30] J. C. Bezdek, R. Ehrlich, and W. Full, "Fcm: The fuzzy c-means clustering algorithm," *Computers & Geosciences*, vol. 10, no. 2, pp. 191–203, 1984.

[31] D. Gustafson and W. Kessel, "Fuzzy clustering with a fuzzy covariance matrix," *IEEE Conference on Decision and Control*, vol. 2, 1979.

[32] I. Gath and A. B. Geva, "Unsupervised optimal fuzzy clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 773–780, 1989.

[33] F. Klawonn, R. Kruse, and H. Timm, *Fuzzy shell cluster analysis*. Springer, 1997.

[34] R. Krishnapuram, H. Frigui, and O. Nasraoui, "Fuzzy and possibilistic shell clustering algorithms and their application to boundary detection and surface approximation," *IEEE Transactions on Fuzzy Systems*, vol. 3, no. 1, pp. 29–43, 1995.

[35] R. Dave and S. Bhamidipati, "Application of the fuzzy-shell clustering algorithm to recognize circular shapes in digital images," in *Proceedings of the International Fuzzy Systems Association Congress*, 1989, pp. 238–24.

[36] R. N. Dave, "Fuzzy shell-clustering and applications to circle detection in digital images," *International Journal Of General System*, vol. 16, no. 4, pp. 343–355, 1990.

[37] R. N. Dave and K. J. Patel, "Fuzzy ellipsoidal shell clustering algorithm and detection of elliptical shapes," in *Proceeding of SPIE Conference Intelligent Robots and Computer Vision IX: Algorithms and Techniques*. International Society for Optics and Photonics, 1991, pp. 320–333.

[38] R. Krishnapuram, H. Frigui, and O. Nasraoui, "The fuzzy c quadric shell clustering algorithm and the detection of second-degree curves," *Pattern Recognition Letters*, vol. 14, no. 7, pp. 545–552, 1993.

[39] D. L. Pham and J. L. Prince, "Adaptive fuzzy segmentation of magnetic resonance images," *IEEE Transactions on Medical Imaging*, vol. 18, no. 9, pp. 737–752, 1999.

[40] M. N. Ahmed, S. M. Yamany, N. Mohamed, A. A. Farag, and T. Moriarty, "A modified fuzzy c-means algorithm for bias field estimation and segmentation of mri data," *IEEE Transactions on Medical Imaging*, vol. 21, no. 3, pp. 193–199, 2002.

[41] S. Chen and D. Zhang, "Robust image segmentation using fcm with spatial constraints based on new kernel-induced distance measure," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 34, no. 4, pp. 1907–1916, 2004.

[42] S. Delbo, P. Gamba, and D. Roccato, "A fuzzy shell clustering approach to recognize hyperbolic signatures in subsurface radar images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 38, no. 3, pp. 1447–1451, 2000.

[43] T. Glenn, A. Zare, and P. Gader, "Bayesian fuzzy clustering," *IEEE Transactions on Fuzzy Systems*, vol. 23, no. 5, pp. 1545–1561, 2015.

[44] E. Airoldi, D. Blei, E. Erosheva, and S. Fienberg, *Handbook of Mixed Membership Models and their Applications*. Chapman and Hall/CRC, 2014.

[45] E. Erosheva, S. Fienberg, and J. Lafferty, "Mixed-membership models of scientific publications," *Proceedings of the National Academy of Sciences*, vol. 101, no. suppl 1, pp. 5220–5227, 2004.

[46] D. Blei, "Mixed membership models," *Princeton University*, 2011.

[47] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother, "A comparative study of energy minimization methods for markov random fields," in *European Conference on Computer Vision*. Springer, 2006, pp. 16–29.

[48] A. Criminisi, "Microsoft research cambridge object recognition image database (version 1.0 and 2.0), 2004."

[49] M. P. Hayes and P. T. Gough, "Synthetic aperture sonar: a review of current status," *IEEE Journal of Oceanic Engineering*, vol. 34, no. 3, pp. 207–224, 2009.

[50] A. Lyons, D. Abraham, and S. Johnson, "Modeling the effect of seafloor ripples on synthetic aperture sonar speckle statistics," *IEEE Journal of Oceanic Engineering*, vol. 35, no. 2, pp. 242–249, 2010.

[51] R. E. Hansen, "Introduction to synthetic aperture sonar," in *Sonar Systems*, N. Z. Kolev, Ed. Triangle Park, NC, USA: Intech, 2011.

[52] J. T. Cobb, "Sonar image modeling for texture discrimination and classification," Ph.D. dissertation, University of Florida, Gainesville, FL, 2011.

[53] J. T. Cobb and J. Principe, "Seabed segmentation in synthetic aperture sonar images," in *Proceedings of SPIE Defense, Security, and Sensing*. International Society for Optics and Photonics, 2011, pp. 80 170M–80 170M.

[54] G. Dobeck, "Algorithm fusion for automated sea mine detection and classification," in *Proceeding of MTS/IEEE Oceans Conference*, vol. 1, 2001, pp. 130–134.

[55] G. J. Dobeck and J. T. Cobb, "Fusion of multiple quadratic penalty function support vector machines (qpfsvm) for automated sea mine detection and classification," in *Proceedings of SPIE*, 2002, pp. 401–411.

[56] J. T. Cobb and A. Zare, "Multi-image texton selection for sonar image seabed co-segmentation," in *SPIE Defense, Security, and Sensing*. International Society for Optics and Photonics, 2013, pp. 87 090H–87 090H.

[57] C. Chen, A. Zare, and J. T. Cobb, "Sand ripple characterization using an extended synthetic aperture sonar model and parallel sampling method," *IEEE Transaction on Geoscience and Remote Sensing.*, vol. 53, no. 10, pp. 5547–5559, 2015.

[58] J. Winn, A. Criminisi, and T. Minka, "Object categorization by learned universal visual dictionary," in *IEEE International Conference on Computer Vision*, 2005, pp. 1800–1807.

[59] D. Li and M. Becchi, "Deploying graph algorithms on gpus: An adaptive solution," in *IEEE International Symposium on Parallel & Distributed Processing*, 2013, pp. 1013–1024.

[60] D. Li, K. Sajjapongse, H. Truong, G. Conant, and M. Becchi, "A distributed cpu-gpu framework for pairwise alignments on large-scale sequence datasets," in *IEEE International Conference on Application-Specific Systems, Architectures and Processors*, 2013, pp. 329–338.

[61] D. Li, S. Chakradhar, and M. Becchi, "Grapid: A compilation and runtime framework for rapid prototyping of graph applications on many-core processors," in *IEEE International Conference on Parallel and Distributed Systems*, 2014, pp. 174–182.

[62] D. R. Jackson and M. Richardson, "High-frequency seafloor acoustics." New York: Springer-Verlag, 2007.

[63] K. Mackenzie, "Bottom reverberation for 530-and 1030-cps sound in deep water," *Journal of Acoustical Society of America*, vol. 33, p. 1498, 1961.

[64] J. W. Caruthers and J. C. Novarini, "Modeling bistatic bottom scattering strength including a forward scatter lobe," *IEEE Journal of Oceanic Engineering*, vol. 18, no. 2, pp. 100–107, 1993.

[65] D. R. Jackson, "Progress and research issues in high-frequency seafloor scattering," in *AIP Conference Proceedings*, vol. 728, 2004, p. 125.

[66] S. Chib and E. Greenberg, "Understanding the metropolis-hastings algorithm," *The American Statistician*, vol. 49, no. 4, pp. 327–335, 1995.

[67] W. K. Hastings, "Monte carlo sampling methods using markov chains and their applications," *Biometrika*, vol. 57, no. 1, pp. 97–109, 1970.

[68] A. F. Smith and G. O. Roberts, "Bayesian computation via the gibbs sampler and related markov chain monte carlo methods," *Journal of the Royal Statistical Society: Series B (Methodological)*, pp. 3–23, 1993.

[69] V. Gopal, "Techniques of parallelization in markov chain monte carlo methods," Ph.D. dissertation, University of Florida, Gainesville, FL, 2011.

[70] P. Mykland, L. Tierney, and B. Yu, "Regeneration in markov chain samplers," *Journal of the American Statistical Association*, vol. 90, no. 429, pp. 233–241, 1995.

[71] R. O. Duda and P. E. Hart, "Use of the hough transformation to detect lines and curves in pictures," *Communications of the ACM*, vol. 15, no. 1, pp. 11–15, 1972.

[72] R. A. Cheel and A. E. Hay, "Cross-ripple patterns and wave directional spectra," *Journal of Geophysical Research: Oceans*, vol. 113, no. C10, 2008.

[73] I. Maier and A. E. Hay, "Occurrence and orientation of anorbital ripples in near-shore sands," *Journal of Geophysical Research: Earth Surface*, vol. 114, no. F4, 2009.

[74] A. E. Johnson and M. Hebert, "Seafloor map generation for autonomous underwater vehicle navigation," *Autonomous Robots*, vol. 3, no. 2-3, pp. 145–168, 1996.

[75] E. Durá, J. Bell, and D. Lane, "Reconstruction of textured seafloors from side-scan

sonar images," in *IEE Proceedings Radar, Sonar and Navigation*, vol. 151, no. 2. IET, 2004, pp. 114–126.

[76] D. Jackson, "Models for scattering from the sea bed," *Proceedings-Institute Of Acoustics*, vol. 16, pp. 161–172, 1994.

[77] D. Langer and M. Hebert, "Building qualitative elevation maps from side scan sonar data for autonomous underwater navigation," in *Proceedings of IEEE International Conference on Robotics and Automation*, 1991, pp. 2478–2483.

[78] R. Zhang, P.-S. Tsai, J. E. Cryer, and M. Shah, "Shape-from-shading: a survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 8, pp. 690–706, 1999.

[79] J. Thomas, W. Kober, and F. Leberl, "Multiple image sar shape-from-shading," *Photogrammetric Engineering and Remote Sensing*, vol. 57, no. 1, pp. 51–59, 1991.

[80] J. Besag, "On the statistical analysis of dirty pictures," *Journal of the Royal Statistical Society*, vol. 48, no. 3, pp. 259–302, 1986.

[81] S. Z. Li and S. Singh, *Markov random field modeling in image analysis*. Springer, 2009, vol. 3.

[82] S. Geman and D. Geman, "Stochastic relaxation, gibbs distributions, and the bayesian restoration of images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 6, pp. 721–741, 1984.

[83] S. Z. Li, *Markov random field modeling in computer vision*. Springer-Verlag New York, Inc., 1995.

[84] I. Dryden, L. Ippoliti, and L. Romagnoli, "Adjusted maximum likelihood and pseudo-likelihood estimation for noisy gaussian markov random fields," *Journal of Computational and Graphical Statistics*, vol. 11, no. 2, 2002.

[85] J. Besag, "Statistical analysis of non-lattice data," *The Statistician*, pp. 179–195, 1975.

[86] ——, "Efficiency of pseudolikelihood estimation for simple gaussian fields," *Biometrika*, pp. 616–618, 1977.

[87] T. Burr and A. Skurikhin, "Pseudo-likelihood inference for gaussian markov random fields," *Statistics Research Letters*, vol. 2, no. 3, 2013.

[88] S. Foucher, M. Germain, J.-M. Boucher, and G. B. Benie, "Multisource classification using icm and dempster-shafer theory," *IEEE Transactions on Instrumentation and Measurement*, vol. 51, no. 2, pp. 277–281, 2002.

[89] J. Besag, "Spatial interaction and the statistical analysis of lattice systems," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 192–236, 1974.

VITA

Chao Chen was born in November 1984, in Shandong, China. She received the Bachelor and Master degrees in Electrical Engineering from Xidian University in July 2007 and May 2010, respectively. She joined the Lane Department of Computer Science and Electrical Engineering at West Virginia University as a Ph.D. student in August 2010 and transfered to the Department of Electrical and Computer Engineering at the University of Missouri in January 2013.