MACHINE LEARNING IN THE OPEN WORLD

by

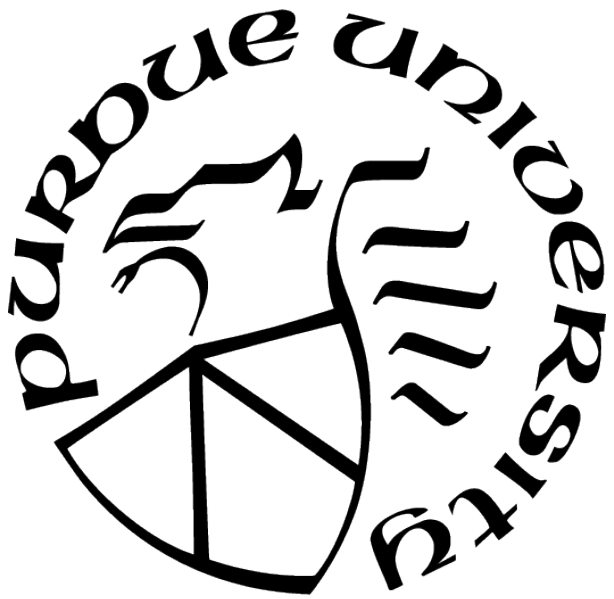
Yicheng Cheng

A Dissertation

Submitted to the Faculty of Purdue University

In Partial Fulfillment of the Requirements for the degree of

Doctor of Philosophy



Department of Computer Science

West Lafayette, Indiana

August 2021

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF COMMITTEE APPROVAL

Dr. Mehmet Murat Dundar, Chair

School of Computer & Information Science

Dr. Petros Drineas

School of Computer Science

Dr. George Mohler

School of Computer & Information Science

Dr. Antonio Bianchi

School of Computer Science

Approved by:

Dr. Kihong Park

ACKNOWLEDGMENTS

Throughout the writing of this dissertation I have received a great deal of support and assistance.

I would first like to thank my supervisor, Prof. Murat Dundar for the continuous support of my PhD study and research, for his expertise and advise in formulating the research questions and methodology. His guidance helped me all the time of research and writing of this thesis. He has taught me to systematically organizing my research and experiments, and gave me lots of valuable suggestions on writing and representation. The methodology and philosophy I learnt from him is beneficial not only for my PhD research but also for my future career.

I would like to acknowledge Prof. George Mohler and Prof. Bartek Rajwa for the collaboration of research papers. These papers are important contributions to my research and thesis. They are publishable only with your collaboration and help.

I would like to thank my advisory committee, Prof. Petros Drineas and Prof. George Mohler for their support and advise throughout the proposal until the defense of my thesis; and my exam committee Prof. Jianzhu Ma and Prof. Antonio Bianchi. Thanks for your time and patience for listening and advising on my research. Your support is valuable for the completion of my research and thesis.

I would also like to thank TL. Jingzheng Qin, Dr. Zhen Qin, and Dr. Haifeng Gong for hosting my internship. Thank you for offering me the opportunity and project for summer 2019 and 2018 at Google. I have gained valuable experiences from my internship projects. Though not included in my thesis, the knowledge I learned is inspiring for my research and thesis.

In addition, I would like to thank my parents and friends for the financial, spiritual, and emotional support. Thanks for their support of my decision of pursuing a PhD in a country on the opposite side of the earth to my hometown; thanks for them to cheer me on when I was frustrated during my research; thanks for my friends who discuss my research ideas with me and gave me inspirations. Only with their support I can finish my PhD.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

By Machine Learning in the Open World, we are trying to build models that can be used in a more realistic setting where there could always be something "unknown" happening. Beyond the traditional machine learning tasks such as classification and segmentation where all classes are predefined, we are dealing with the challenges from newly emerged classes, irrelevant classes, outliers, and class imbalance. At the beginning, we focus on the Non-Exhaustive Learning (NEL) problem from a statistical aspect. By NEL, we assume that our training classes are non-exhaustive, where the testing data could contain unknown classes. And we aim to build models that could simultaneously perform classification and class discovery. We proposed a non-parametric Bayesian model that learns some hyper-parameters from both training and discovered classes (which is empty at the beginning), then infer the label partitioning under the guidance of the learned hyper-parameters, and repeat the above procedure until convergence. After obtaining good results on applications with plain and low dimensional data such flow-cytometry and some benchmark datasets, we move forward to Non-Exhaustive Feature Learning (NEFL). For NEFL, we extend our work with deep learning techniques to learn representations on datasets with complex structural and spatial correlations. We proposed a metric learning approach to learn a feature space with good discrimination on both training classes and generalize well on unknown classes. Then we developed some variants of this metric learning algorithm to deal with outliers and irrelevant classes. We applied our final model to applications such as open world image classification, image segmentation, and SRS hyperspectral image segmentation and obtained promising results. Finally, we did some explorations with Out of Distribution detection (OOD) to detect irrelevant sample and outliers to complete the story.

1. INTRODUCTION

Modern machine learning techniques have reached very high standards on recognition tasks such as image category recognition, text classification, spectral recognition etc. Machines can now outperform humans even in domains where years of training and education is required to achieve desired level of domain expertise. However, most of their success relies heavily on manual supervision, a crucial aspect of learning a robust mapping from observations to labels and that usually requires a large number of representative samples for satisfactory performance. Traditional machine learning can only learn categories predefined by manual labels and are ill-fitted for the open world settings, where new categories may continuously emerge and data obscured by noise and outliers introduce additional challenges for machine learning.

Compared with a closed world setting in which all categories are predefined before training, open world setting aligns better with real world scenarios where there could always be something "unknown" happening. Moreover, majority of the supervised learning tasks assume that class sizes are balanced, whereas in the non-stationary real world environment frequency of categories usually follow a heavy-tailed distribution. A few dominated classes covers most of the population with some small classes scattered around.

The characteristic of non-stationary environment clearly exacerbates the challenge of learning in the open world. In order to deal with emerging categories, noise and outliers in the presence of rare classes, we need the model to learn high level concepts rather than just the mathematical mapping from observations to labels. Instead of mimicking human supervisions we want the model to "understand" the underlying mechanism that forms categories based on its "past experiences" with similar categories. For example if a person sees some fruits he/she has never seen before, he/she can easily tell which of them belong to same species and which are not based on the common sense acquired from past experience. In other words, we wish the ideal machine learning model to implicitly learn the so called "common sense" from its past experience. Learning "common sense" seems to be trivial to human beings, but it is proved to be extremely challenging for machines as it requires learning abstract knowledge at different levels of abstraction [1], [2].

Research areas such as zero-shot learning [3], few-shot learning [4], transfer learning [5], out of distribution detection [6], open-set classification [7] each study different but related aspects of the closed-world assumption impeding traditional classification. However, none of these methods study novel class discovery problem when there is no information about emerging classes. For example, zero-shot learning recognize new classes based on their descriptive attributes instead of using an extensive set of training data-label pairs, while few-shot learning reduces the number of training samples for new categories to a "few" shots to save the effort of labeling massive training samples. Few shots learning achieves this by learning to transfer knowledge from classes with massive labeled samples available over to those that exist only with few samples. Out-of-distribution detection aims to recognize samples that doesn't belong to any of the known classes. In contrast open-set classification build classifiers that can operate in the presence of out-of-distribution samples with the goal of classifying only samples of known classes while ignoring out-of-distribution samples.

Related lines of work that study novel class discovery are learning to cluster [8] and transfer clustering [9]. Although they both address novel class discovery, these methods separate learning from known categories and achieve novel category discovery in two independent stages. First they pretrain a feature extractor or a metric function using labeled data and then perform unsupervised fine-tuning on the target datasets based on the learned feature extractor/metric function. No portion of the labeled data is used in the fine-tuning stage and no portion of the unlabeled data is used in the pretraining stage. They also assume that no samples of known classes will appear in testing, which is a restriction that doesn't hold for most real world scenarios where emerging samples could originate from both known and unknown classes.

Our research combines recognition of known classes with discovery of unknown classes in a single framework to have the following core functionalities. First, samples from existing classes can be classified with acceptable accuracy as in traditional supervised learning. Second, samples from emerging classes can be identified with acceptable sensitivity and specificity even when multiple emerging classes are present in the data. Third, the classifier model should be flexible enough to differentiate not only between existing and emerging classes but among different emerging classes as well.

This is the so called non-exhaustive learning (NEL) paradigm first studied for a bio-detection problem in [10], [11]. In non-exhaustive learning we assume that the training classes is non-exhaustively listed, i.e., new classes may emerge after training. Our initial attempt for NEL was the implementation of the adaptive I^2GMM (AI^2GMM) model [12], which is a two layer hierarchical non-parametric Bayesian model with adaptive hyper-parameters based on I^2GMM [13]. In AI^2GMM, we assume that class centers and covariances are drawn from a Normal and Inverse Wishart (NIW) bivariate distribution. The hyperparameters of this distribution are estimated by maximizing sample posterior likelihood conditioned on current class assignments. By constraining samples of known classes to remain within their own classes during inference, the inferred hyper-parameters are expected to capture "the past experience" from known classes to guide the generation of emerging classes. In order to have more flexibility over cluster shapes and to be able to model non-Gaussian distributions, AI^2GMM models each class in the form of an infinite mixture of Gaussian components. The component means are sampled from a distribution controlled by their class center and their scaled class covariance, while all components inheriting the same class share the same class covariance. Details of the AI^2GMM can be found in 2.1.

Although AI^2GMM achieves good performance on applications involving well-separated classes in low dimensional feature space such as cell type discovery in flow-cytometry and some multi-spectral benchmark datasets [12], its applicability to complex data with structural information is limited. Structural data that involve images and texts not only contain high-level abstract features but are also susceptible to high levels of noise and are often obscured by outliers and other unwanted artifacts. To accommodate structural information in our models we employ feature learning techniques before applying non-exhaustive learning algorithms. Feature learning techniques such as stacked auto-encoder [14], generative adversarial networks [15], and self supervised learning methods are known for their success in visual feature extraction. However, unsupervised feature learning does not guarantee learning class discriminatory features, which are essential for non-exhaustive learning. Thus, we aim to learn features that could discriminate between known classes but also generalize well on unknown classes. In other words we want to implement a non-exhaustive feature learning

framework that can learn to better separate unknown classes by implicitly learning the high level concepts relevant for class separation from known classes.

We explore different ideas for non-exhaustive feature learning in chapter 3. First we use variational Infinite Gaussian Mixture Model (IGMM) (see section 3.1 for details), which takes an IGMM as a prior and learns the auto-encoder by minimizing the KL divergence. This method turns out to be very sensitive to initial assignments. We believe that the interleaved training of both features and class assignments hampers effective training as the initial class compositions obtained from under-trained features cannot capture actual class distributions well. This is a common issue with algorithms that adopt interleaved learning of features and cluster distributions [9], [16]. An intuitive way to solve this problem is to first pretrain feature extraction networks to generate good quality features that could achieve high clustering accuracy, and then fine tune the model with the learned clusters as described in [9], [16]. These works that utilize unsupervised feature learning methods have shown to be ineffective for non-exhaustive learning. If we pretrain the feature extractors in a fully unsupervised fashion then the information from the labeled data cannot be incorporated into learning. On the other hand if we pretrain the feature extractors to maximize separation of known classes the model will be biased towards known classes and generalize poorly on unseen classes.

Instead of directly maximizing the likelihood of known classes, we investigate methods that indirectly use labels so that the feature extractor learned on known classes could still generalize well on unknown classes. Inspired by the triplet loss idea we propose a method called Centroid Margin Loss (CML). It is based on the idea that the distance between a sample to the closest class center except its own class should be greater than to its own class center at least by a margin. We provide two versions of this loss function, one for labeled data, another one for unlabeled data. Our preliminary results suggest robust generalization can be achieved on unknown classes even when the feature extractor is merely trained by labeled data. Details can be found in section 3.2. We adjusted the original CML into two variants, Asymmetric CML (ACML) and Kernel-ed CML (KCML) to handle the challenge of the existence of irrelevant classes and outliers/tagging-errors. We then applied the variants on two applications, one is non-exhaustive semantic cell segmentation for elec-

tron microscopy images 3.4, another is the SRS hyper-spectral image segmentation 3.3. Our experiments showed the effective of CML on learning a discriminative feature without losing the generalization for unknown classes.

In addition to non exhaustive learning, we also studied out-of-distribution detection for detecting outliers to build models that could operate in open world settings. Distance-based methods are popular in the literature for performing outlier detection [17], [18]. Inspired by the mahalanobis distance based out-of-distribution detection technique introduced in [19] we aim to integrate outlier detection with nonexhaustive learning to be able to perform recognition, class discovery, and outlier detection in a single unified framework as future work.

1.1 Background and Motivation

Traditional recognition problems can be categorized into supervised classification and unsupervised clustering. Our work originated from the need to classify bacterial pathogens in a non-stationary environment where new types of bacteria can continuously emerge due to high mutation rate [20]. It is impossible to build an exhaustive training dataset to include past and future types of bacteria, while fully unsupervised clustering can not produce a classification accuracy comparable to that of supervised classification. Non-exhaustive learning problem is introduced with the aim of achieving as good of a performance as supervised classification on known classes and can generalize the supervised information of known classes for better clustering unknown clusters.

Motivated by the CRISM mineral discovery application [21] we then extend the non-exhaustive learning into the open world learning problem due to a need for simultaneously classifying known classes, discover novel classes, and detect outliers. CRISM is a hyperspectral image dataset of Mars with massive amount of data, and there may exist minerals that do not have matching lab spectra, and it is noisy due to aging instrument, poor device calibration, atmospheric effects, distance etc. It is not possible to build an exhaustive training dataset unless all of the more than 20,000 image cubes are examined by planetary scientists, which makes it meaningless to use machine learning to aid the mineral discovery. Moreover,

in addition to hyperspectral images of Mars, similar needs are expected to arise in the near future for analyzing hyperspectral images of Jupiter moon Titan. The need for simultaneous outlier/noise detection is arising from the complex composition of additive and multiplicative noise. The distribution of noise depends on class distributions and also is image and region specific, which makes denoising challenging.

2. NON EXHAUSTIVE LEARNING

Non-Exhaustive Learning (NEL) lies in the center of our research topic, and is the most important and the most challenge task for machine learning in the open world. As described in chapter 1, by Non-Exhaustive Learning, we are trying to perform unified classification and class discovery without the assumption that the training class is exhaustively defined. Except perform NEL in a unified way, this could also be achieved by separating the classification and clustering, for example doing an open set classification then perform clustering on rejected samples. But instead of performing the two tasks separately, we believe that we can improve the class discovery performance by learning the underlying class generation mechanism from known classes, and further aid the classification performance of known classes by extending the dataset with unlabeled testing samples.

In this chapter, We first analyze and try to solve this problem from a statistical aspect. See section 2.1 for the details. Section 2.2 is an application of this model. Based on the statistical model, we also include another line of thought in section 2.3.

2.1 Adaptive I^{2}GMM: A Unified Approach to Classification and Clustering for Non-exhaustive Learning

Our first attempt for non-exhaustive learning is the adaptive I^{2}GMM model [12], which is built on top of a highly flexible two-layer non-parametric Gaussian mixture model (I^{2}GMM) [13] and unifies classification with clustering to perform classification, class discovery, modeling, and recovery all at once.

The first work for NEL uses a Gaussian mixture model and continuously augments existing set of Gaussian components by creating a new one each time the maximum of class conditional likelihood values for an incoming sample is found to be below a designated threshold [10]. A more systematic solution was later offered in [11], [22] based on the infinite Gaussian mixture model (IGMM) that eliminates the need for thresholding likelihood values and uses the concentration parameter of the Dirichlet process to encode prior knowledge about probability of new classes. These works uses training data available from the initial set of classes for initializing a partially observed IGMM model and estimating its hyperparameters.

However, the performance of these techniques can significantly vary from one problem to other depending on how well each class data can be captured by a single Gaussian distribution. Although IGMM offers some flexibility toward more accurate estimation of the probability density function by generating an arbitrarily large number of Gaussian components, it is well known that more accurate estimation of the overall data density does not necessarily lead to more accurate estimation of individual class distributions especially when classes emerge with skewed and multi-modal distributions. Under such settings IGMM creates a large number of extraneous classes which may severely restrict the utility of IGMM for non-exhaustive learning and yield poor classification and clustering performance.

In the clustering domain this model mismatch problem was recently tackled by I^{2}GMM [13]. In I^{2}GMM the lower layer estimates the density of the overall dataset by clustering individual data points to components, while the upper layer associates components with clusters to allow for cluster recovery. More specifically, the generative model uses a two-layer hierarchical Dirichlet process mixture (DPM) model where the lower layer uses one DPM for each cluster and the upper layer uses a global DPM for modeling cluster shapes and sizes. Thanks to its two-layer nonparametric structure, I^{2}GMM can model a wide spectrum of distributions and has proved very effective in clustering datasets with multi-modal and skewed clusters.

I^{2}GMM has several hyperparameters that offer great flexibility for encoding domain knowledge into the model. If these hyperparameters can be tuned to capture different aspects of the underlying data model I^{2}GMM can become an effective model for non-exhaustive learning as well. Both labeled and unlabeled data, even when they originate from different subsets of classes, can be effectively blended through a bilateral inference that restricts assignment of labeled points to a subset of components associated with their class of origin while assigning all unlabeled points to any of the existing components without any restrictions. With this approach the model can adapt to unlabeled data by creating new classes and/or by creating new components for existing classes.

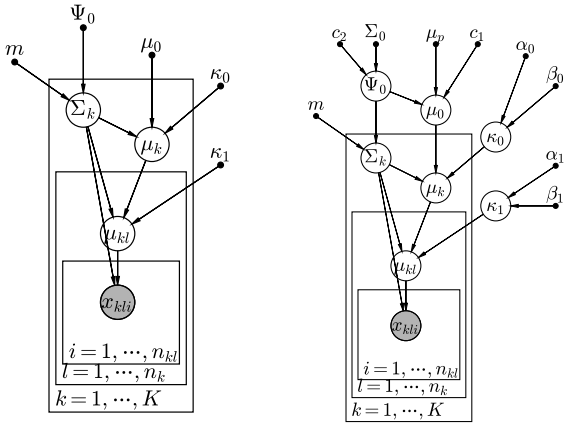
The success of this restricted inference strategy mostly depends on how effectively the fixed values assigned to hyperparameters can capture the complexity of the underlying data model. If the values are optimized to fit the training data better the model becomes too

restrictive and generates redundant components and/or classes. On the contrary when the values are chosen too vague the model becomes too general and generates too few or no new classes. Choosing a set of values that offers an acceptable trade off between the model being too restrictive versus too general is not easy unless the underlying application domain and data model are well known.

In non-exhaustive learning two types of discrepancies can occur between labeled and unlabeled data. The first source of discrepancy arises when an unlabeled data point originates from a yet unobserved component of a known class with a multi-mode distribution or from the tail end of a skewed distribution. The second source of discrepancy arises when an unlabeled data point originates from a yet unobserved class. Thus, apart from model flexibility the most critical aspect of non-exhaustive learning is the reconciliation of these discrepancies between labeled and unlabeled data by recovering as much information as possible about the underlying data model. I^2GMM has hyperparameters that control the shape of components, the scattering of class centers around the data mean, the scattering of component centers around their class means, as well as class and component sizes. These parameters define the underlying data model. Thus, the compromise between labeled and unlabeled data can be more easily reached if the model is modified to allow for learning of these hyperparameters using both labeled and unlabeled data as opposed to using fixed values as in standard I^2GMM. To avoid unlabeled data dominating this learning process we impose restrictions on the inference. Unlike unlabeled data points which can freely switch component and class memberships during inference, labeled data points are only allowed to switch component membership. This way labeled data points are constrained to remain in their class of origin. For inference we use a collapsed Gibbs sampler for inferring class and component indicator variables, which are sampled conditioned on the point estimates of hyperparameters.

2.1.1 The Infinite Mixture of Infinite Gaussian Mixtures (I^2GMM)

In a Gaussian Mixture Model (GMM) each cluster is represented by a single Gaussian distribution characterized by its mean and covariance. In the finite GMM the number of Gaussian components is fixed. The infinite version of GMM (IGMM) is obtained by defining



(a) The generative model for I^2GMM

(b) The generative model for AI^2GMM

Figure 2.1. The probabilistic graphical model (PGM) for I^2GMM and AI^2GMM

a Dirichlet process (DP) prior over the components. The DP prior serves as a distribution over the Gaussian distributions. Its base distribution acts as a Bayesian prior over the mean vectors and covariance matrices while its concentration parameter models the number of components and their sizes. IGMM is an improvement over its finite version as the number of components can be inferred directly from the data. However, the core modeling aspect of GMM does not change with IGMM as each cluster data is still modeled by a single Gaussian component. This creates a problem when fitting IGMM onto dataset with multi-mode and skewed cluster distributions. In an attempt to more accurately estimate density of the overall data IGMM generates multiple Gaussian components for such clusters. However, more accurate density estimation does not necessarily translate into improved clustering performance with IGMM due to inherent one-to-one association imposed between components and clusters.

I^{2}GMM is introduced to address this limitation of IGMM. I^{2}GMM models each cluster by IGMM and creates dependency across all clusters by using a global DP to model the base distributions of local DPs. This two layer architecture of I^{2}GMM allows for modeling of non-Gaussian cluster shapes because each cluster data can be modeled by an arbitrarily large number of components in the lower layer. The global DP in the upper layer establishes the association between components and clusters, which also allows for information sharing across clusters and their components.

The generative model for I^{2}GMM is given by

$$\begin{aligned} H &= NIW(\mu, \Sigma|\mu_0, \Psi_0, \kappa_0, m) \\ &= N(\mu|\mu_0, \kappa_0^{-1}\Sigma)W^{-1}(\Sigma|\Psi_0, m) \\ G &\sim DP(\gamma H) \\ (\mu_k, \Sigma_k) = \theta_k &\sim G \\ H_k &= N(\mu_k, \kappa_1^{-1}\Sigma_k) \\ G_k &\sim DP(\alpha H_k) \\ \mu_{kl} &\sim G_k \\ x_{kl_i} &\sim N(\mu_{kl}, \Sigma_k) \end{aligned} \tag{2.1}$$

The generative model is also illustrated by a plate diagram in Figure 2.1a. Based on this generative model, a global Dirichlet Process is defined with base distribution H and concentration parameter γ. H is a bivariate Normal $\times$ Inverse Wishart distribution (NIW) with hyperparameters $\{\mu_0, \Psi_0, \kappa_0, m\}$. To generate data points, we first draw a discrete mixing measure G from $DP(\gamma H)$. Then the cluster centers μ_k and covariances Σ_k are sampled from G. Next, we define a local DPM with base distribution H_k and concentration parameter α for each cluster generated by the global DPM, where H_k is defined by a Gaussian centered at μ_k with covariance $\kappa_1^{-1}\Sigma_k$. Then a cluster specific discrete mixing measure G_k is drawn from the local DPM and a component with its mean vector drawn from G_k is generated. All components generated this way share the same cluster specific covariance matrix Σ_k. Finally data points x_{kl_i} in cluster k and component l are generated from the component with center μ_{kl} and covariance Σ_k.

Inference is performed by a collapsed Gibbs sampler similar to [23]. After integrating out hidden variables $\mu_{kl}, \mu_k, \Sigma_k$, we can sample both cluster and component indicators from the posterior predictive distributions defined by sufficient statistics and hyperparameters. The posterior predictive distribution of a data point x belonging to component s and cluster k is given by (2.2). Detailed derivations are available in [13].

$$
\begin{aligned}
P(x, t = s, c = k | X, T, C) &= T(x | \bar{\mu}_s, \bar{\Sigma}_s, \bar{v}_s) \\
\bar{\mu}_s &= \frac{n_{ks}\bar{x}_{ks} + \bar{\kappa}_s\bar{\mu}_k}{n_{ks} + \bar{\kappa}_s} \\
\bar{\Sigma}_s &= \frac{\Sigma_0 + \sum_{l:c_l=k} S_{kl} + S_{ks} + S_\mu}{\frac{(n_{ks}+\bar{\kappa}_s)\bar{v}_s}{n_{ks}+\bar{\kappa}_s+1}} \\
\bar{\mu}_k &= \frac{\sum_{l:c_l=k} \frac{n_{kl}\kappa_1}{n_{kl}+\kappa_1}\bar{x}_{kl} + \kappa_0\mu_0}{\sum_{l:c_l=k} \frac{n_{kl}\kappa_1}{n_{kl}+\kappa_1} + \kappa_0} \\
\bar{\kappa}_s &= \frac{\left(\sum_{l:c_l=k} \frac{n_{kl}\kappa_1}{n_{kl}+\kappa_1} + \kappa_0\right)\kappa_1}{\sum_{l:c_l=k} \frac{n_{kl}\kappa_1}{n_{kl}+\kappa_1} + \kappa_0 + \kappa_1} \\
S_\mu &= \frac{n_{ks}\bar{\kappa}_s}{\bar{\kappa}_s + n_{ks}}(\bar{x}_{ks} - \bar{\mu}_k)(\bar{x}_{ks} - \bar{\mu}_k)^T \\
\bar{v}_s &= m + \sum_{l:c_l=k} (n_{kl} - 1) + n_{ks} - d + 1
\end{aligned}
\tag{2.2}
$$

where $X = \{x_{\mathrm{i}}\}_{\mathrm{i}=1}^{n}$ is the set of n data points; $T = \{t_{\mathrm{i}}\}_{\mathrm{i}=1}^{n}$ are the component indicator variables for data points, $C = \{c_{\mathrm{i}}\}_{\mathrm{i}=1}^{L}$ are cluster indicator variables for components; $T(x|\bar{\mu}_s, \bar{\Sigma}_s, \bar{v}_s)$ denotes the student-t distribution with mean $\bar{\mu}_s$, covariance $\bar{\Sigma}_s$, and degree of freedom $\bar{v}_s$; $\bar{x}_{ks}$ and $S_{ks} = \sum_{\mathrm{i}=1}^{n_{ks}} (x_{kl_{\mathrm{i}}} - \bar{x}_{ks})(x_{kl_{\mathrm{i}}} - \bar{x}_{ks})^T$ are the sample mean and scatter matrix for component s in cluster k; n_{ks} is the number of data points in component s of cluster k. The latent variables in T and C are sampled one at a time conditioned on all other variables.

Given initial values for T and C the inference is performed by first sampling the component indicator variable for each data point. The posterior distribution for sampling the component indicator variable t_{i} for a data point x_{i} in class k is given by

$$P(t_{\mathrm{i}}|X, T, C) \propto \begin{cases} n_{kl} T(x_{\mathrm{i}}|\bar{\mu}_s, \bar{\Sigma}_s, \bar{v}_s), & t_{\mathrm{i}} = l \\ \alpha T(x_{\mathrm{i}}|\bar{\mu}_*, \bar{\Sigma}_*, \bar{v}_*), & t_{\mathrm{i}} = L_k + 1 \end{cases} \tag{2.3}$$

where L_k is the number of components in cluster k; the subscript $*$ is used to indicate parameters for a new component, which are obtained by dropping the sufficient statistics for the selected component in equation (2.2). Note that these samplings are conditioned on the cluster indicator variables C. As a result t_{i} is sampled within its own cluster k.

After sampling all component indicators for data points we then sample cluster indicators for each component l by the posterior distribution

$$P(c_l|X, T, C) \propto \begin{cases} L_k \prod_{\mathrm{i}:t_{\mathrm{i}}=l} T(x_{l\mathrm{i}}|\bar{\mu}_s, \bar{\Sigma}_s, \bar{v}_s), & c_l = k \\ \gamma \prod_{\mathrm{i}:t_{\mathrm{i}}=l} T(x_{l\mathrm{i}}|\bar{\mu}_*, \bar{\Sigma}_*, \bar{v}_*), & c_l = K + 1 \end{cases} \tag{2.4}$$

where K is the number of clusters. These two steps define a single Gibbs sweep. Depending on the size of the data convergence to the target distribution may require up to a few thousand Gibbs sweeps.

2.1.2 Adaptive I^{2}GMM

I^{2}GMM offers great flexibility for modeling dataset with an unknown number of classes where classes can have continuous arbitrary distributions. However, tuning or optimizing the values of the hyperparameters based on a limited set of labeled data compromises this flexibility and yields a model that may not fit unlabeled data well.

In the Bayesian context it is a common practice to distribute uncertainty surrounding hyperparameters across multiple layers by treating hyperparameters as variables. An additional layer makes the model less sensitive to changes in the values of the hyperparameters. However, in a purely unsupervised setting such a strategy significantly expands the state space and convergence to the target distribution becomes more of a challenge. In the non-exhaustive setting presence of labeled data may help eliminate a significant portion of the potential modes the sampler can converge. However, if a large number of classes are missing and/or labeled data from existing classes are not representative of their underlying distributions some of the most promising modes can get eliminated as well if the labeled dataset is given too much emphasis during model inference. In this section we discuss the formulation of an adaptive I^{2}GMM that is designed as a trade off between model being too flexible yet uninformative vs. too restrictive and unaccommodating.

Toward achieving this end we first modify the generative model of I^{2}GMM by creating an additional layer in the Bayesian hierarchy that treats the most data sensitive hyperparameters of the model (μ_0,Ψ_0,κ_0, κ_1) as variables. Then, we implement an implicit Bayesian model averaging strategy to infer hyperparameters without sacrificing much from model flexibility. Finally, we infer class indicator variables for unlabeled data and component indicator variables for both labeled and unlabeled data by a restricted Gibbs sampler that also takes into account potential overlaps across classes in the feature space. We define a Normal prior over μ_0, a Wishart prior over Ψ_0, Gamma priors over κ_0 and κ_1. These prior distributions are chosen for conjugacy.

$$\begin{aligned} &\mu_0 \sim N\left(\mu_p, (\Psi_0 c_1)^{-1}\right),\ \Psi_0 \sim W(\Sigma_0, c_2), \\ &\kappa_0 \sim \Gamma\left(\alpha_0, \beta_0\right),\ \kappa_1 \sim \Gamma\left(\alpha_1, \beta_1\right) \end{aligned} \tag{2.5}$$

With these additions the plate diagram of the generative model for AI^{2}GMM is shown in Figure 2.1b.

Hyperparameter Inference

Note that there is no closed form solution for the posterior predictive distributions of the hyperparameters. We cannot sample them as we sample class and component indicator variables by a Gibbs sampler. We resort to point estimation techniques for these hyperparameters as described below.

Estimating Ψ_0 and μ_0

Ψ_0 is the hyperparameter that defines the expected shape of components whereas μ_0 is the hyperparameter that encodes expected mean vector for the individual class mean vectors. If we estimate these parameters to maximize their corresponding posterior distributions conditioned on a specific configuration of data points and estimations of other parameters then their estimates become highly sensitive to this configuration. If this configuration is not a good representation of the underlying data model the inference becomes more prone to getting stuck at a poor local optima. To maintain model flexibility it is more reasonable to obtain version of these estimates that are configuration independent. This can be done by model averaging through maximizing the joint likelihood of all data points originating from a new class.

The joint likelihood of the data originating from a new class is

$$p(X|\mu_0, \Psi_0) = \prod_{\mathrm{i}}^{n} p(x_{\mathrm{i}}|\mu_0, \Psi_0) \tag{2.6}$$

The posterior for Ψ_0 is

$$p(\Psi_0|X) = W^{-1}(\Psi_0|\Sigma_0, c_2) \prod_{\mathrm{i}}^{n} p(x_{\mathrm{i}}|\mu_0, \Psi_0) \tag{2.7}$$

where

$$p(x_{\mathrm{i}}|\mu_0, \Psi_0) = St(x_{\mathrm{i}}|\mu_0, \frac{m+1-D}{1+\kappa_0^{-1}+\kappa_1^{-1}}\Psi_0^{-1}, m-d+1) \overset{\text{Laplace appr.}}{\approx} N(x_{\mathrm{i}}|\mu_0, b\Psi_0) \tag{2.8}$$

where $b = \frac{1+\kappa_0^{-1}+\kappa_1^{-1}}{m+1}$. We use Laplace approximation to approximate Student-t distribution by a Normal distribution so that a closed-form estimate for Ψ_0 can be obtained. We substitute equation (2.8) back into (2.7) to get

$$p(\Psi_0|X) = W^{-1}(\Psi_0|\Sigma_0, c_2)\prod_{\mathrm{i}=1}^{n} N(x_{\mathrm{i}}|\mu_0, b\Psi_0) \tag{2.9}$$

We can derive the estimates for Ψ_0 and μ_0 by maximizing the posterior

$$\hat{\Psi}_0 = \frac{\Sigma_0 + b^{-1}\sum_{\mathrm{i}=1}^{n}(x_{\mathrm{i}}-\mu_0)(x_{\mathrm{i}}-\mu_0)^T}{n+c_2+D+1} \tag{2.10}$$

$$\hat{\mu}_0 = \frac{2bc_1\mu_p + \sum_{\mathrm{i}}^{n} x_{\mathrm{i}}}{2bc_1+n} \tag{2.11}$$

Estimating κ_0 and κ_1

κ_0 is the hyperparameter that adjusts the separation among class means. The smaller the κ_0, the farther apart the class centers will be from the center of the data, i.e., μ_0. κ_1 is the hyperparameter that adjusts the separation among component means of a given class. The smaller the κ_1, the farther apart the component centers will be from their corresponding class center, and classes will tend to have multi-modal shapes. According to the plate diagram shown in Figure 2.1b, the posterior for these hyperparameters conditioned on the point estimates of hidden variables are given in (2.12). We use the $\hat{}$ notation to distinguish estimates from variables. We also use $*$ to indicate conditioning on the current configuration of the data points and estimates of all other hidden variables.

$$\begin{aligned} p(\kappa_1|*) &\propto Ga(\kappa_1|\alpha_1, \beta_1)\prod_{k\in C}\prod_{l:c_l=k} N(\hat{\mu}_{kl}|\hat{\mu}_k, \hat{\Sigma}_k\kappa_1^{-1}) \\ p(\kappa_0|*) &\propto Ga(\kappa_0|\alpha_0, \beta_0)\prod_{k\in C} N(\hat{\mu}_k|\hat{\mu}_0, \hat{\Sigma}_k\kappa_0^{-1}) \end{aligned} \tag{2.12}$$

The point estimates of the hyperparameters are obtained by maximizing the corresponding posterior distributions.

$$\begin{aligned}\hat{\kappa}_0 &= \frac{2(\alpha_0 - 1) + dK}{2\beta_0 + \sum_k (\hat{\mu}_k - \hat{\mu}_0)^T \hat{\Sigma}_k^{-1} (\hat{\mu}_k - \hat{\mu}_0)} \\ \hat{\kappa}_1 &= \frac{2(\alpha_1 - 1) + d\sum_k n_k}{2\beta_1 + \sum_k \sum_{l:c_l=k} (\hat{\mu}_{kl} - \hat{\mu}_k)^T \hat{\Sigma}_k^{-1} (\hat{\mu}_{kl} - \hat{\mu}_k)}\end{aligned} \tag{2.13}$$

Note that these point estimates also depends on the hidden variables μ_{kl}, μ_k, Σ_k, which we estimate as follows. The posterior for these variables conditioned on the point estimates of hyperparameters and other hidden variables are given in (2.14).

$$\begin{aligned}p(\Sigma_k|*) &\propto IW(\Sigma_k|\hat{\Psi}_0, m) N(\hat{\mu}_k|\hat{\mu}_0, \Sigma_k \hat{\kappa}_0^{-1}) \\ &\qquad \prod_{l:c_l=k} N(\hat{\mu}_{kl}|\hat{\mu}_k, \Sigma_k \hat{\kappa}_1^{-1}) \prod_{l:c_l=k} W(S_{kl}|\Sigma_k, n_{kl} - 1) \\ p(\mu_k|*) &\propto N(\mu_k|\hat{\mu}_0, \hat{\Sigma}_k \hat{\kappa}_0) \prod_{l:c_l=k} N(\hat{\mu}_{kl}|\mu_k, \hat{\Sigma}_k \hat{\kappa}_1^{-1}) \\ p(\mu_{kl}|*) &\propto N(\mu_{kl}|\hat{\mu}_k, \hat{\Sigma}_k \hat{\kappa}_1^{-1}) N(\bar{x}_{kl}|\mu_{kl}, \frac{\hat{\Sigma}_k}{n_{kl}})\end{aligned} \tag{2.14}$$

The point estimates of these hidden variables are obtained my maximizing their corresponding posterior distributions.

$$\begin{aligned}\hat{\Sigma}_k &= \frac{\hat{\Psi}_0 + \hat{\kappa}_0 (\hat{\mu}_k - \hat{\mu}_0)(\hat{\mu}_k - \hat{\mu}_0)^T + S_k + \sum_{l:c_l=k} S_{kl}}{m + d + 2 + \sum_{l:c_l=k} n_{kl}} \\ S_k &= \hat{\kappa}_1 \sum_{l:c_l=k} (\hat{\mu}_{kl} - \hat{\mu}_k)(\hat{\mu}_{kl} - \hat{\mu}_k)^T \\ \hat{\mu}_k &= \frac{\hat{\mu}_0 \hat{\kappa}_0 + \sum_l \hat{\mu}_{kl} \hat{\kappa}_1}{\hat{\kappa}_0 + n_k \hat{\kappa}_1} \\ \hat{\mu}_{kl} &= \frac{\hat{\mu}_k \hat{\kappa}_1 + \bar{x}_{kl} n_{kl}}{\hat{\kappa}_1 + n_{kl}}\end{aligned} \tag{2.15}$$

Sampling class and component indicators by a restricted Gibbs sampler

Given an unlabeled dataset X_u along with a labeled dataset X_ℓ and its corresponding label set C_ℓ, a restricted Gibbs sampler is implemented to preserve the class composition of the labeled data during inference. For the points in the unlabeled dataset both the class

indicators C_u and component indicators T_u are unknown and both need to be inferred. For the data points in the labeled set class indicators C_ℓ are available but component indicators T_ℓ are unknown and need to be inferred.

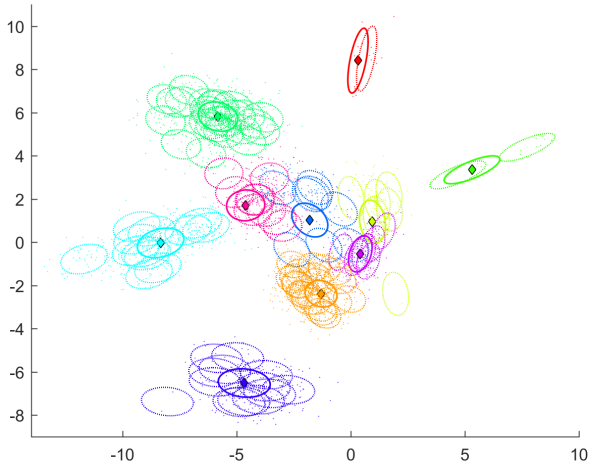
As the class labels for labeled data points are available, a straightforward restriction would be to assign a component containing labeled data points to its corresponding class without considering other classes. Although such an approach could be effective for datasets with well separated class distributions, it creates additional problems when classes overlap or some of the classes exhibit heavy-tailed distributions. Assigning a component located in a region of the feature space that overlaps with other class distributions to one of the observed classes severely limits the modeling capacity of AI^2GMM. Many unlabeled data points from unobserved classes, which so happen to be assigned to the same component as labeled data, would be incorrectly assigned to the same observed class if such a restriction were to be imposed.

We tackle this potential class overlap problem by executing a preinference stage where we infer component indicators for all labeled data points and identify a fixed proportion of them from each class as potential outliers based on their class conditional likelihood values. During actual inference each Gibbs scan includes hyperparameter estimation followed by sampling of component and class indicators, respectively. Component indicators for each data point is sampled without any restrictions according to (2.3) with the hyperparameters replaced by their then-current point estimates where $X = X_u \cup X_\ell$, $T = T_u \cup T_\ell$, and $C = C_u \cup C_\ell$. When sampling class indicators for each component we impose a restriction as follows. All components contain non-outlier labeled points are assigned to their respective classes without sampling. On the other hand all components composed of only unlabeled points and outlier labeled points are assigned to classes according to (2.4). Finally, to preserve class composition, all outlier labeled points are assigned back to their respective classes.

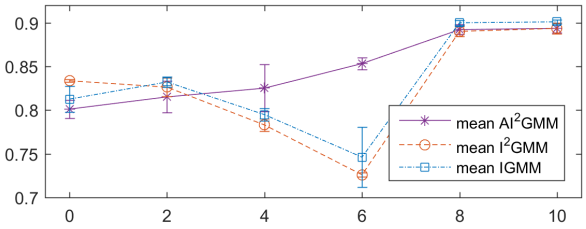
This restricted Gibbs sampler not only preserves class composition of labeled data points but also allow for joint classification and clustering of unlabeled data points. Three possible scenarios exist for unlabeled data points: perform standard classification for unlabeled data points share the same component assigned to one of the observed classes with labeled data

points; perform classification with new component of an observed class discovered for unlabeled data points in a component assigned to one of the observed classes with no labeled data points; perform class discovery for unlabeled data in component assigned to new class.

2.1.3 Experiments



(a) Scatter plot of the simulated data. Each class data is shown by a different color. Contours with solid lines indicate class-specific distributions. Contours with dashed lines indicate component distributions.



(b) Mean F_1 scores as a function of the number of observed classes in the labeled data.

Figure 2.2. Simulated data experiment comparing AI^2GMM, I^2GMM, and $IGMM$.

We perform experiments with simulated and benchmark datasets to validate the performance of AI^{2}GMM for non-exhaustive learning. For each dataset we set aside twenty percent of all available points as labeled data. However, we do not use the labels in the labeled dataset all at once. To create a partially observed labeled dataset we start with zero observed classes, i.e., no labels are used, and gradually increase the number of observed classes by two until all of the available classes for each dataset are represented in the labeled dataset. The case with no observed classes in the labeled data corresponds to fully unsupervised setting, i.e., clustering, whereas the case with all classes observed corresponds to fully supervised setting, i.e., classification. All cases in between are considered non-exhaustive learning.

We compare the performance of AI^{2}GMM against IGMM and I^{2}GMM. IGMM and I^{2}GMM have been mainly used in the literature for unsupervised learning problems. Here, we use a restricted Gibbs sampler scheme to adapt these models for non-exhaustive learning. In IGMM labeled points are assigned to their class of origin without sampling to preserve class composition for all observed classes. For I^{2}GMM we use the same restricted sampling strategy we adopt for AI^{2}GMM. In other words we introduced two new versions of I^{2}GMM for non-exhaustive learning that both use the same restricted Gibbs sampler we discussed in Section 2.1.2 but differ in the way hyperparameters are treated. I^{2}GMM fixes hyperparameters to vague values whereas AI^{2}GMM dynamically estimates them using both labeled and unlabeled data.

All features in each dataset are normalized to have zero mean and unit variance. Fixed sets of vague hyperparameters are used for both IGMM and I^{2}GMM. For IGMM: $\mu_0 = \mathbf{0}$, $\Psi_0 = \mathbf{I}$, $\kappa_0 = 0.1$, $m = d + 2$. For I^{2}GMM: $\mu_0 = \mathbf{0}$, $\Psi_0 = \mathbf{I}$, $\kappa_0 = 0.1$, $\kappa_1 = 0.5$, $m = d + 2$ where d is the number of features, $\alpha = 1$, $\gamma = 1$ are used for all datasets. For AI^{2}GMM we consider 20% of labeled data from each observed class as outliers and select vague values for $c_1 = 0.1$, $c_2 = d + 2$, $\beta_0 = 20$, $\alpha_0 = 3$,$\beta_1 = 20$, $\alpha_1 = 11$. The number of Gibbs sweeps is set to 1000 in all three algorithms. For the preinference stage implemented to infer components of the labeled data in AI^{2}GMM we used 200 Gibbs sweeps.

We use the mean F_1 score to evaluate performance for all algorithms, which is computed as described below. Given the ground truth labels and predicted labels for the unlabeled data we build a confusion matrix C of K rows and L columns, where K is the number of ground truth classes and L is the number of predicted classes. We assume that rows 1 through M of C are corresponding to the M observed classes and columns 1 through M are corresponding to predicted classes. Mean F_1 score is computed as $F_1 = \frac{1}{K}\sum_{k=1}^{K} F_1^k$ where F_1^k is evaluated differently for observed and unobserved classes

$$F_1^k = \begin{cases} \dfrac{2c_{kk}}{\sum_{\mathrm{i}=1}^{K} c_{\mathrm{i}k} + \sum_{\mathrm{j}=1}^{L} c_{k\mathrm{j}}} \text{if k is observed} \\ \dfrac{2\max\left(c_{kM+1}, ..., c_{kL}\right)}{\sum_{\mathrm{i}=1}^{K} c_{\mathrm{i}l} + \sum_{\mathrm{j}=1}^{L} c_{k\mathrm{j}}} \text{if k is unobserved} \end{cases} \tag{2.16}$$

where c_{ij} is the element for the ith row and jth column of C and l is the index returned by max. Each experiment is repeated five times and average F_1 scores are reported in all experiments to account for the stochastic nature of inference.

Experiments with Simulated Data

We use a two layer Gaussian mixture model similar to I^{2}GMM to generate a 2D simulated data with multi-mode class distributions. For illustration purposes we fixed the number of classes to 10, the total number of components to 100, and the total number of points to 3000. We sample the number of components within each class and the number of data points within each component according to a Dirichlet prior with parameters set to γ and α, respectively. Class centers μ_k and covariance matrices Σ_k are sampled from a NIW with $\mu_0 = \mathbf{0}$, $\Psi_0 = \mathbf{I}$, $\kappa_0 = 0.03$, $m = 4$. Component centers μ_{kl} are sampled from a Normal distribution with mean μ_k and covariance matrix $\kappa_1^{-1}\Sigma_k$ with $\kappa_1 = 0.5$. Finally data points for each component are sampled from a Normal distribution with mean μ_{kl} and covariance matrix Σ_k. The generated dataset is shown in Figure 2.2a. Classes with varying sizes exhibit multi-mode patterns with highly unbalanced component sizes, which makes this a challenging dataset for clustering.

Experiments with Benchmark Datasets

We used several benchmark datasets from different domains to further validate the proposed algorithm. *FLC1* is a 12-channel multispectral airborne image containing several land cover types most of which are known to have multi-mode class distributions. *NDDg1* and *stemcell* are two of the flow cytometry datasets used in the FlowCAP I competition [24]. Both datasets exhibit skewed class distributions. We also selected three datasets (*letter*, *sat*, and *uci_har*) from UCI Machine Learning repository. These are all datasets with continuous valued features but with various numbers of classes and dimensions as shown in Table 2.1. The dimensionality of uci_har is reduced to 30 by PCA. Experiments with all other datasets are performed with the original dimensionality.

Table 2.1. F1 scores for benchmark datasets

dataset	# observed classes	IGMM	I^{2}GMM	AI^{2}GMM
NDDg1 $d = 12$ $n = 44620$	0	0.502	**0.696**	0.691
	2	0.464	0.654	**0.663**
	4	0.497	0.699	**0.727**
	6	0.535	0.816	**0.825**
	7	0.527	0.788	**0.830**
FLC1 $d = 12$ $n = 69500$	0	0.776	**0.777**	0.774
	2	0.758	0.788	**0.797**
	4	0.797	0.773	**0.838**
	6	0.825	0.872	**0.894**
	9	0.822	0.931	**0.934**
letter $d = 16$ $n = 20000$	0	0.416	**0.474**	0.409
	5	0.399	0.311	**0.461**
	7	0.379	0.286	**0.495**
	13	0.431	0.427	**0.575**
	26	0.591	**0.861**	0.840
sat $d = 36$ $n = 6435$	0	0.523	0.494	**0.603**
	2	**0.579**	0.396	0.459
	4	0.632	0.452	**0.660**
	6	0.764	0.812	**0.814**
stemcell $d = 6$ $n = 9936$	0	0.514	0.651	**0.773**
	2	0.487	0.868	**0.930**
	4	0.566	**0.901**	0.890
uci_har $d = 561$ $n = 10299$	0	0.614	**0.689**	0.614
	2	0.565	0.730	**0.790**
	4	0.822	**0.863**	0.834
	6	0.883	**0.928**	0.910

Discussion

The following observations can be made from the results of experiments performed with both simulated and benchmark datasets. First, there is a strong trend that favors I^{2}GMM and AI^{2}GMM over IGMM under all experimental settings, which justifies the need for using more flexible models for modeling class distributions. Second, in a fully unsupervised setting (rows with second column equal to zero in Table 2.1) vague values assigned to hyperparameters can be more useful than values estimated from data. When no observed classes are available treating all hyperparameters as variables enlarges the state space and makes

AI^{2}GMM more susceptible to a poor local optima problem. Third, in non-exhaustive settings estimating hyperparameters helps improve the performance of AI^{2}GMM significantly. However as the results of experiments with simulated data as well as on *sat* and *letter* benchmark datasets suggest the performance of I^{2}GMM is less predictable under non-exhaustive settings. Restricting the Gibbs sampler while the model is constrained by a fixed set of vague hyperparameters creates clustering configurations that do not conform well with the constrained model in I^{2}GMM. Under these constrained settings the model favors existing classes over new ones. This effect is more evident when the number of observed classes is small. When most or all of the classes are observed I^{2}GMM can still achieve moderately high F_1 without generating new classes and thus the effect of the model's limitation becomes negligible on the overall performance.

2.2 Applications to Seismic Fault Detection by a Coupled ETAS-I^{2}GMM Point Process

In addition to our non-exhaustive learning research, we have extended the unsupervised I^{2}GMM mentioned in 2.1.1 to a seismic fault detection application and published as [25].

The epidemic-type aftershock sequence (ETAS) model of earthquake occurrence [26], [27] is a self-exciting point-process model where the conditional intensity $\lambda(t, x, y|H_t)$ of events is determined by a stationary Poisson intensity generating spontaneous earthquake events along with a dynamic term representing a branching process of aftershocks:

$$\lambda(t, x, y|H_t) = \mu(x, y) + \sum_{\mathrm{i}:t_\mathrm{i}<t} g(t - t_\mathrm{i}, x - x_\mathrm{i}, y - y_\mathrm{i}, m_\mathrm{i}). \tag{2.17}$$

Here (x, y) is the epicenter of an earthquake event described by longitude and latitude in decimal degrees, m is its magnitude on the Richter scale computed using a body-wave magnitude formula [28], $H_t = \{(t_\mathrm{i}, x_\mathrm{i}, y_\mathrm{i}, m_\mathrm{i}) : t_\mathrm{i} < t\}$ is the history of all earthquake events up to time t in a catalog, and $\mu(x, y)$ is the background intensity reflecting spatial heterogeneity of spontaneous earthquakes and the fact that earthquake catalogs with aftershocks removed are approximately Poisson in time [29].

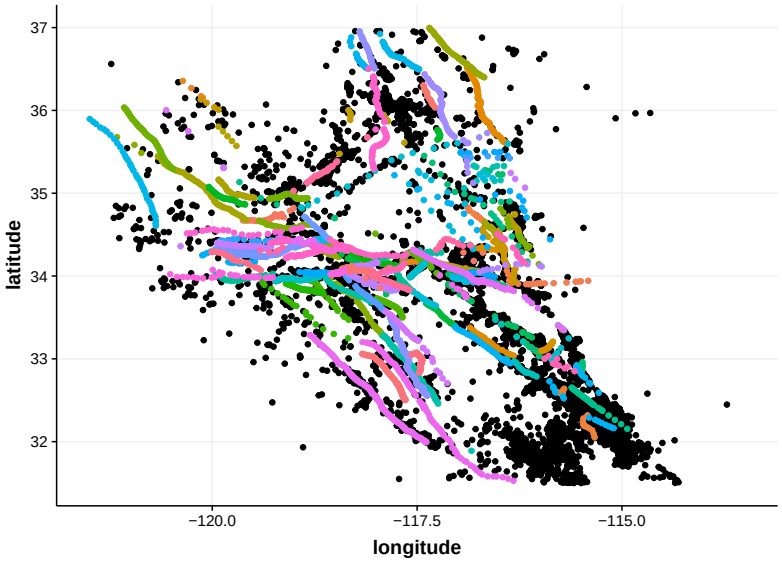


Figure 2.3. Southern California earthquakes magnitude 2.5 and greater (black) and faults corresponding to the Community Fault Model 3.0 (marked by lines).

The space-time-magnitude distribution of parent-offspring events in the branching process given by the function $g(t, x, y, m)$ is called the triggering kernel, typically following Omori's law [30]:

$$g(t - t_{\mathrm{i}}, x - x_{\mathrm{i}}, y - y_{\mathrm{i}}, m_{\mathrm{i}}) = \frac{K_0 \mathrm{e}^{a(m_{\mathrm{i}} - m_0)}}{(t - t_{\mathrm{i}} + c)^{(1+\omega)}((x - x_{\mathrm{i}})^2 + (y - y_{\mathrm{i}})^2 + d)^{(1+\rho)}} \tag{2.18}$$

where m_0 is the cutoff magnitude of the dataset under study [26] and $(K_0, a, c, \omega, d, \rho) > 0$ are parameters to be estimated. Estimation of Equation 2.17 typically consists of constructing a non-parametric estimate for $\mu(x, y)$ along with finding maximum-likelihood estimators for the parameters of the triggering kernel in Equation 2.18. Methods for maximizing the likelihood include quasi-Newton [26] and expectation-maximization (EM) [31], and the most common estimators for $\mu(x, y)$ are spatial histograms [31], [32] or isotropic kernel density estimators [33], [34].

Earthquakes cluster at multiple scales, as earthquakes cluster locally through aftershock activity but also over larger scales along fault lines (see Figure 2.3). While there is research on the reconstruction of aftershock clusters from event data [33], [35], existing point-process models of earthquake activity fail to capture spatial clustering patterns at the larger scale of fault lines. In particular, histograms and kernel density estimators are able to capture spatial heterogeneity in the risk of spontaneous earthquakes, but the methods capture variation over only one scale. To our knowledge, our work here is the first to attempt to reconstruct the community fault model [36] with a statistical model based on earthquake event data.

In this paper we introduce a new type of ETAS model that can capture multiscale clustering in earthquake patterns. In particular, we propose using an infinite mixture of infinite Gaussian mixtures (I^2GMM) [13] to estimate the background rate of earthquakes $\mu(x, y)$. For each spatial cluster, the I^2GMM uses a different Dirichlet process mixture of Gaussians (DPMG) that simultaneously predicts the number of clusters along with performing model inference. While I^2GMM has been introduced for high-dimensional clustering and ETAS is well known in seismology, what is new in this paper is the use of I^2GMM for modeling the intensity of a point process and the coupling of these two techniques for multiscale modeling of space-time event patterns. Through the use of an expectation-maximization algorithm, the benefit of our approach is that earthquakes are assigned membership to aftershock clusters in addition to a larger scale fault line cluster.

Another advantage of our approach is that multi-modal and skewed spatial clusters are more accurately captured. In the case of spatial earthquake patterns, each fault may be considered as a separate cluster with multi-modality and skewness that the I^2GMM can handle better than histograms and KDE estimators. In particular, a fault is represented in the background rate of spontaneous earthquakes by I^2GMM as a cluster of several Gaussians. Unlike kernel density estimation where each kernel corresponds to one event in the dataset at which it is centered, each Gaussian in a spatial cluster of I^2GMM is not necessarily centered at a historical event and can generate multiple spontaneous earthquakes in the future. One additional advantage of the I^2GMM model is that earthquakes are assigned membership to spatial clusters inferred by the model. In this research, we explore the relevance of

spatial cluster membership to automatic detection of fault lines within the ETAS-I^{2}GMM framework.

2.2.1 Methods

Infinite mixture of infinite Gaussian mixtures

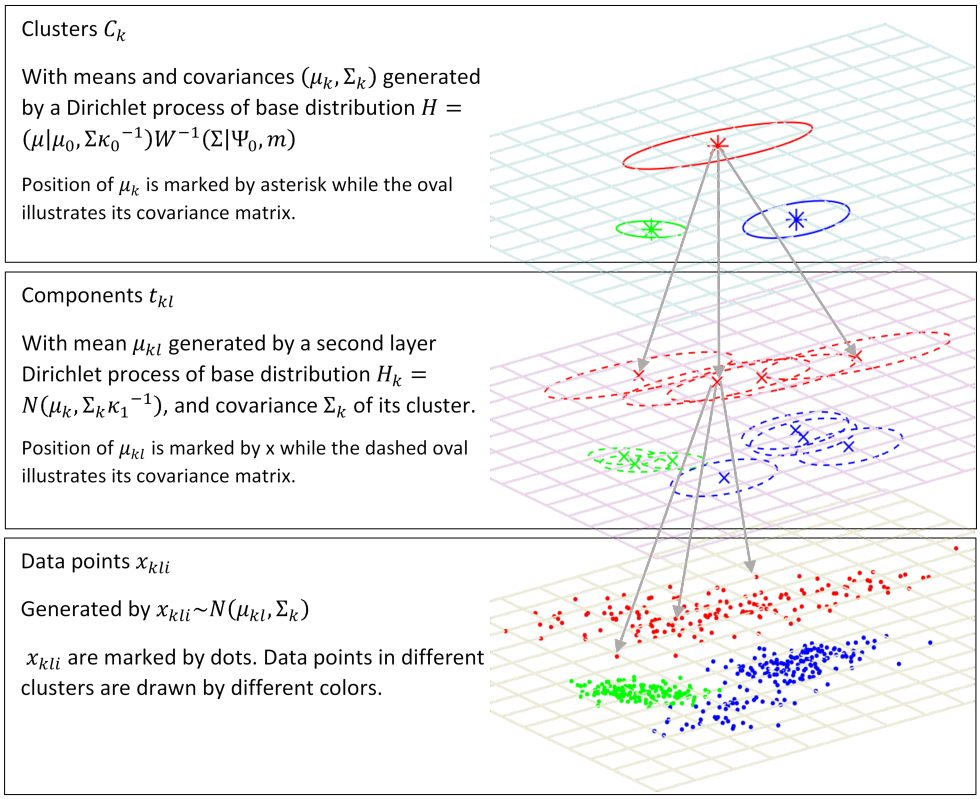


Figure 2.4. The hierarchy of I^{2}GMM model illustrated on a synthetic dataset.

Since we have already gave the details of I^{2}GMM in 2.1.1, we will not repeat it here. We believe that I^{2}GMM has three unique features that make it very suitable for the estimation of background intensity $\mu(x, y)$ in the ETAS model.

- As a two-layer non-parametric model, I^{2}GMM allows the number of clusters and the number of mixture components in each cluster to grow arbitrarily large, offering great flexibility in modeling clusters with multi-mode/skewed distributions. This is the main

feature that distinguishes I^{2}GMM from other model-based clustering techniques that use one component for each cluster.

- As a Bayesian model, I^{2}GMM has hyper-parameters that can be tuned to recover clusters with varying shapes and different levels of rarity without facing singularities during model estimation. This distinguishes I^{2}GMM from purely data-driven techniques such as finite mixture of Gaussians and t-distributions that rely on EM and its extensions during model learning.

- As a hierarchical model, I^{2}GMM can share parameters not only across different clusters but also across different components of the same cluster. In other words, I^{2}GMM assumes that components are generated independently only when conditioned on the unique parameter defining their clusters of origin. This differentiates the proposed work from other techniques that estimate a large number of Gaussian components and merges them sequentially to recover clusters, thus violating component dependence.

In Figure 2.4 we provide an illustration of the generative model for I^{2}GMM. Where t_{kl} indicates the l^{th} component in the k^{th} cluster C_k; x_{kl_i} indicates the ith data point in the l^{th} component in the k^{th} cluster. In the generative process, t_{kl} is a Gaussian distribution and C_k is a Gaussian mixture defined by its components. We will use the top-level label (i.e., C_k) to identify different clusters of spontaneous earthquakes that can be used to predict fault membership of each event. The lower-level labels could also be used to identify faults or sub-faults (where seismologists may want to consider refining their labels); however, in this paper we will restrict our analysis to the top-level labels.

To perform inference with the I^{2}GMM model using spatial event data, we first initialize the cluster and component indicators for each event to some arbitrary values (for example put all data in the same component of a cluster) and then use a collapsed Gibbs sampler to infer values for indicator variables one at a time, given all other indicator variables [13]. Conditioned on the indicator variables, the location vectors and scale matrices are determined by maximizing the complete data log-likelihood and have closed-form solutions. One sweep of the Gibbs sampler will go over all events in the dataset; convergence typically requires several hundred to thousand Gibbs sweeps.

EM inference for ETAS

The ETAS model given in Equation 2.17 can be viewed as a branching process where spontaneous events occur according to a Poisson process with intensity $\mu(x, y)$. Events (from all generations) give birth to direct offspring events determined by the triggering kernel $g(t - t_{\mathrm{i}}, x - x_{\mathrm{i}}, y - y_{\mathrm{i}}, m_{\mathrm{i}})$.

Given an initial guess for the parameters of the triggering kernel in Equation 2.18 and the background rate $\mu(x, y)$, the branching structure along with the model parameters of the triggering kernel can be estimated using an EM algorithm [31], [37]. In the E-step of the EM algorithm the probability p_{ij} that event i is a direct offspring of event j is estimated, along with the probability p_{i}^{b} that the event was generated by the Poisson process with rate $\mu(x_{\mathrm{i}}, y_{\mathrm{i}})$.

$$p_{\mathrm{ij}} = \frac{g(t_{\mathrm{i}} - t_{\mathrm{j}}, x_{\mathrm{i}} - x_{\mathrm{j}}, y_{\mathrm{i}} - y_{\mathrm{j}})}{\lambda(t_{\mathrm{i}}, x_{\mathrm{i}}, y_{\mathrm{i}})}, \tag{2.19}$$

$$p_{\mathrm{i}}^{b} = \frac{\mu(x_{\mathrm{i}}, y_{\mathrm{i}})}{\lambda(t_{\mathrm{i}}, x_{\mathrm{i}}, y_{\mathrm{i}})}, \tag{2.20}$$

Given the probabilistic estimate of the branching structure, the complete data log-likelihood is then maximized in the M-step (using standard methods for estimating a Pareto distribution) [31], providing an estimate of the model parameters.

Joint inference of the ETAS-I^{2}GMM

We propose three variants for inferring the joint ETAS-I^{2}GMM model.

ETAS-I^{2}GMM 1. In the first variant we start by clustering all events spatially using I^{2}GMM. We then evaluate the convex hull of each cluster and enforce $\mu(x, y)$ to be constant in the convex hull. $\mu(x, y)$ and other parameters are then estimated using the EM algorithm in Section 2.2.1 with the rectangular cells in Section 2.2.1 replaced by the estimated convex hulls.

ETAS-I^{2}GMM 2. In the next variant we perform joint inference using a Monte-Carlo EM algorithm. In particular, at each EM iteration we perform the following steps:

i. (I^{2}GMM-step) Sample background events from probabilistic branching structure p_{ij}. Rather than clustering all events using I^{2}GMM now we cluster the sampled background events only. Estimate $\mu(x, y)$ based on the clusters of background events the same as in ETAS-I^{2}GMM 1.

ii. (E-step) Estimate probabilistic branching structure and model parameters of triggering kernel as in Section 2.2.1.

ETAS-I^{2}GMM 3. In the last variant we use a weighted I^{2}GMM algorithm in place of the i step in variant two above. Instead of estimating $\mu(x, y)$ based on clusters of sampled background events, we estimate $\mu(x, y)$ based on clusters generated by weighted I^{2}GMM on all events whose weights are estimated by (2.20). In the first EM iteration, since p_{i}^b does not exist we initialize the weight to 1.

Baseline ETAS model with histogram estimator

We use the histogram estimator proposed in [31] as a baseline model for comparison. In particular, we let the background rate $\mu(x, y)$ be a constant:

$$\mu(x, y) = \mu_k \text{ if (x,y) is in cell } k, k \in 1, \ldots, K \tag{2.21}$$

over each rectangular cell of a regular grid. There are then $K + 6$ parameters $\theta = (\mu_1, \ldots, \mu_K, a, c, d, w, \rho, K_0)$ that we need to estimate (assuming there are K cells in the grid) and for that purpose we use the EM algorithm in [31].

ETAS model with variable kernel estimates

For a second comparison we use kernel density estimation with variable bandwidth as proposed in [33]. Here $\mu(x, y)$ is estimated as

$$\mu(x, y) = \frac{1}{T} \sum_{\mathrm{j}} p_{\mathrm{j}}^{b} k_{d_{\mathrm{j}}}(x - x_{\mathrm{j}}, y - y_{\mathrm{j}}) \tag{2.22}$$

where T is the time span of all events, p_{j}^{b} is the background probability defined in (2.20), d_{j} is the varying bandwidth calculated for each event j according to the distance of its n_p nearest neighbor, and $k_d(x, y)$ denotes the Gaussian kernel function $\frac{1}{2\pi d} \exp\{-\frac{x^2+y^2}{2d^2}\}$. For all experiments, we set the parameter $n_p = 3$ as suggested by [38]. The rest of the parameters are estimated according to Section 2.2.1 using the same EM algorithm.

2.2.2 Experiments and Results

Experiment 1: goodness of fit of ETAS-I^{2}GMM applied to CA earthquakes 3.5 and greater since 2000

We apply our models to the California earthquake-event data filtered by year (greater than 2000) and magnitude (greater than 3.5). The geographic bounds range from $46.116 >$ latitude > 29.615 and $-113.581 >$ longitude > -130.427. The dataset is divided into training and testing using time point 2010-01-01 00:00:00 as cutoff. All events before this time stamp are placed in the training dataset, while all events after it are placed in the testing dataset. We performed experiments with the following seven models to analyze how the performance varies by adopting different modeling strategies:

1. ETAS-I^{2}GMM 1.
2. ETAS-I^{2}GMM 2.
3. ETAS-I^{2}GMM 3.
4. 4×4 grid baseline model.
5. 3×4 grid baseline model.

6. 3×3 grid baseline model.

7. ETAS-KDE described in Section 2.2.1.

Note that experiments 1 to 3 are repeated 10 times and the means of the likelihoods are recorded. For I^{2}GMM we run 400 Gibbs sweeps; the hyper-parameters are set as follows: $\mu_0 = [36.4603; -119.3265]$ the mean of the dataset; $\Sigma_0 = [21.4972 \quad 0; 0 \quad 23.1351)]$ the diagonal matrix with diagonal entries equal to the latitude-longitude variances of the dataset; $m = 22$; $\kappa_0 = 0.1$; $\kappa_1 = 0.5$. The values of m, κ_0, κ_1 are tuned to let I^{2}GMM generate fewer or equivalent number of clusters as the 4×4 grid baseline model.

We use the log-likelihood function

$$\text{logL} = \sum_{\text{i}=1}^{N} \log(\lambda(t_\text{i}, x_\text{i}, y_\text{i})) - \int_0^T \int_S \lambda(t, x, y) dx dy dt \tag{2.23}$$

to evaluate the competing models for the background intensity. The results are shown in Table 2.2.

Table 2.2. Log-likelihood model comparison.

Model	logL	$\sum_\text{i} \log(\lambda_\text{i})$	$\int \lambda$
ETAS-I^{2}GMM 1	**-4619**	-1206	3413
ETAS-I^{2}GMM 2	-4686	-1283	3403
ETAS-I^{2}GMM 3	-4716	-1319	3397
4×4 Grid	-4980	-1590	3390
3×4 Grid	-4937	-1607	3330
3×3 Grid	-5023	-1582	3440
ETAS-KDE	-4860	-1441	3419

All three ETAS-I^{2}GMM models outperform those with histogram or kernel density estimators. Between the three ETAS-I^{2}GMM variants, the best-performing model is variant 1, where I^{2}GMM is first estimated and the EM algorithm is run separately to estimate the parameters of the triggering kernel. It is worthwhile to note that finer clusters do not necessarily yield better results. Even though the 4×4 grid model generates more clusters it produces lower likelihood than the 3×4 grid model.

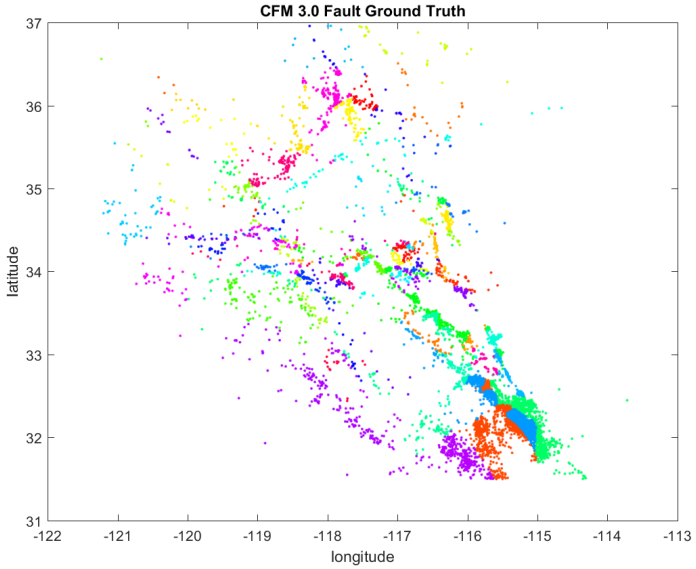


Figure 2.5. Fault line cluster membership using the nearest CFM 3.0 fault to each earthquake (ground truth).

Experiment 2: ETAS-I^{2}GMM for event-fault linkage from space-time event data

Next, we investigate the extent to which the ETAS-I^{2}GMM model can learn fault structure from space-time event data. For this purpose we use the Community Fault Model 3.0 [36], which is a three dimensional representation (latitude, longitude, and elevation) of faults in Southern California. The CFM is a collaborative project undertaken by scientists of the Southern California Earthquake Center (SCEC) for studying active faults and earthquake phenomena and to improve regional earthquake hazard assessments. Our goal here is to assess how well ETAS-I^{2}GMM recovers a 2D projection of the CFM 3.0 using only space-time-magnitude earthquake incident data as input. In particular, we generate a fault label for each event in the dataset by assigning fault membership as the nearest fault in CFM 3.0 (see Figure 2.5).

The ETAS-I^{2}GMM-predicted label is taken from the first layer of the I^{2}GMM model clusters, where offspring events are assigned to the spatial cluster of their nearest neighbor among background events. To allow for comparison to the CFM 3.0, we restrict the

geographic bounds of the CA earthquake event data to $36.958 >$ latitude > 31.518 and $-113.719 >$ longitude > -121.176, but we expand the magnitude threshold down to 2.5.

There are 145 actual fault lines in CFM 3.0, but all the models we used in previous experiments generate at most 26 clusters. For this dataset, we added two additional models in the experiments for fault recovery:

- An I^{2}GMM with parameters tuned to generate approximately 145 clusters on average; this version is named as ETAS-I^{2}GMM 145 in our experiments.
- A 16×15 grid model that contains 143 non-empty clusters.

Table 2.3. Clustering accuracy comparison of fault classification. The log-likelihoods are also included. We are not able to evaluate the accuracy score for ETAS-KDE since it doesn't generate spatial clusters.

	Accuracy	$\overline{Acc}_{10}$	logL	$\sum_i \log(\lambda_i)$	$\int \lambda$
ETAS-I2GMM 145	0.46	0.52	**5495**	16772	11277
ETAS-I2GMM 1	**0.50**	**0.67**	4688	16034	11346
ETAS-I2GMM 2	0.45	0.46	4466	15820	11354
ETAS-I2GMM 3	0.41	0.33	4495	15904	11409
Grid 16x15	0.45	0.47	4384	15703	11319
Grid 4x4	0.37	0.37	4247	15723	11476
Grid 3x4	0.36	0.28	4224	15694	11470
Grid 3x3	0.35	0.32	4198	15673	11475
ETAS-KDE	–	–	4066	15464	11398

Given that the number of clusters estimated by I^{2}GMM may be different from the number of faults in the CFM 3.0, we evaluate the success of fault-cluster recovery by considering the percentage of correctly classified data points. In addition to the overall clustering accuracy, we evaluate the mean clustering accuracy for the 10 largest faults, which contains 67% of data points across 145 faults. In Table 2.3 we present the clustering accuracy for the seven models listed in 2.2.2 as well as the two additional models. To calculate the clustering accuracy, we first align the generated clusters with the ground-truth classes using the Hungarian algorithm [39], [40], and then calculate the percentage of the data points that fall into their classes of origin. We adopt clustering accuracy for its simplicity and its invariance to potential

misalignment between ground truth and predicted class labels. Mean clustering accuracy is calculated as below:

$$\overline{Acc} = \frac{1}{|C^*|} \sum_{C_k^* \in C^*} \frac{|C_k^* \cap C_k|}{|C_k^*|} \tag{2.24}$$

where for each event, C_k^* is fault cluster assignment of event k and C_k is the predicted cluster assignment of event k. Here C^* contains all fault clusters under consideration; C_k is the predicted cluster corresponding to C_k^* after alignment; $C_k^* \cap C_k$ indicates data points in both C_k^* and C_k; $|S|$ denote the cardinality of the set S. To compute the mean clustering accuracy for the ten largest faults $\overline{Acc}_{10}$ we set C^* to contain the ten largest true fault clusters in the above equation.

Here again we see that ETAS-I^{2}GMM 1 performs best both in terms of clustering accuracy and $\overline{Acc}_{10}$. All the ETAS-I2GMM models outperform ETAS with a histogram estimator or a kernel density estimator in terms of likelihood, which is consistent with the results we have in Experiment 1 on the CA earthquake data. In Figure 2.6 we plot the clusters recovered corresponding to ETAS-I^{2}GMM 145, ETAS-I^{2}GMM 1, 16×15 grid, 4×4 grid, and the true clusters for a better understanding of this outcome. Despite the fact that I^{2}GMM generated only 26 unique clusters on average compared to 145 actual fault lines in CFM 3.0, a meaningful clustering accuracy of 0.5 was achieved. Results suggest that a majority of events in fault clusters that tend to have elongated, skewed, and in some cases multi-mode shapes are clustered correctly by I^{2}GMM. In contrast, the clusters in the two histogram models have abrupt boundaries formed from the grid irrespective of the shape of the underlying faults, as shown in Figure 2.6. Moreover, when we adjust the parameters of the I^{2}GMM to get approximately the same number of clusters as the true number of fault clusters, we observe that the clustering accuracy does not improve owing to erroneous splitting of events belonging to larger fault lines into multiple clusters. This is also true for the 16×15 grid model that generates 143 non-empty clusters. Although the clustering accuracy improves with this model compared to the grid model with a smaller number of clusters, overall clustering accuracy achieved by this model is still less than that achieved by I^{2}GMM with 26 clusters (0.45 vs 0.50). The difference in clustering accuracy between

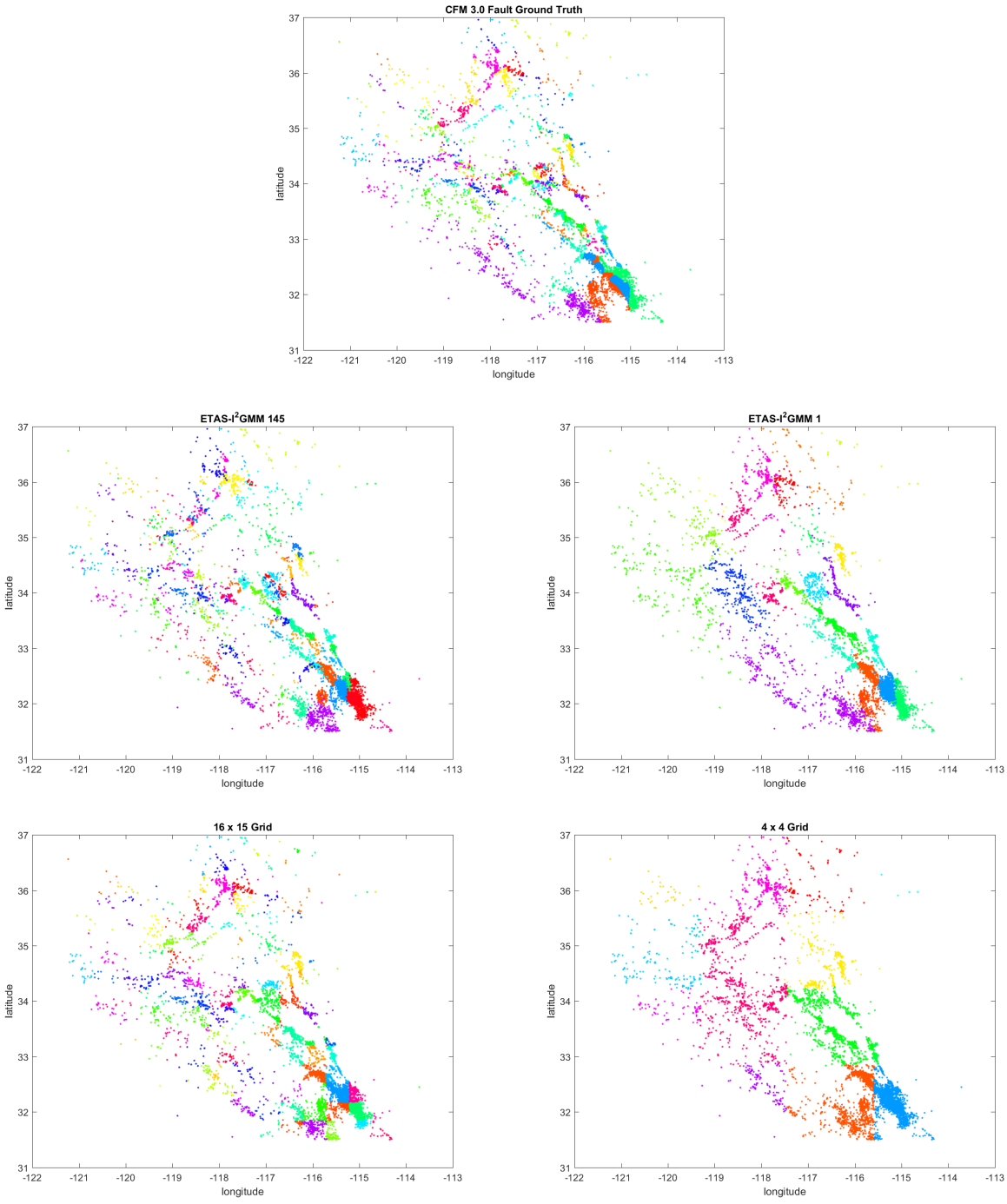


Figure 2.6. True and predicted CFM fault groupings. Events with same labels are shown by the same color.

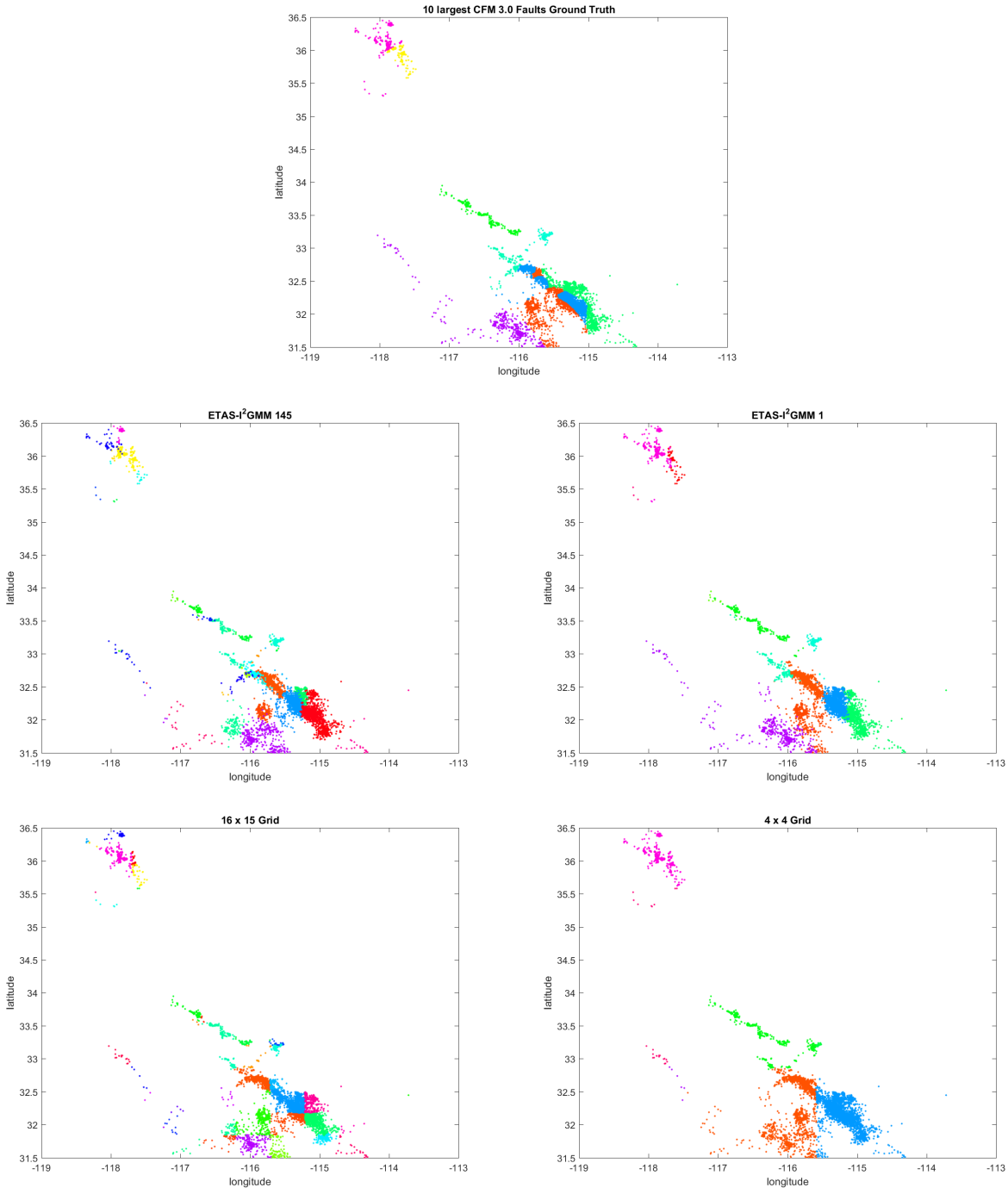


Figure 2.7. Ten largest CFM faults and recovered clusters. Events with same labels are shown by the same color.

the two models increases in favor of I^{2}GMM when we take into account only the largest ten fault clusters (0.47 vs 0.67). This is a natural result of the grid model's arbitrarily splitting fault clusters, compared to the more effective handling of elongated fault cluster shapes by I^{2}GMM.

We illustrate this over-splitting problem in Figure 2.7 by plotting the clustering results of the ten largest faults. From the $\overline{Acc}_{10}$ results in Table 2.3 and Figure 2.7 we can see that ETAS-I^{2}GMM 1 did the best by achieving a mean clustering accuracy of 0.67 across ten faults while recovering several of them by a clustering accuracy of over 0.9. On the other hand, for the grid models the $\overline{Acc}_{10}$ values are consistent with their corresponding overall accuracies.

2.2.3 Discussion

We introduced a coupled ETAS-I^{2}GMM model for jointly estimating multi-scale clustering in earthquake data with parameters governing earthquake productivity and self-excitation. We also introduced what we believe is a novel machine-learning task for statistical seismology, namely estimating CFM fault clusters using unlabeled space-time-magnitude event data. Improving upon algorithms aimed at solving this task could aid in the development of future versions of CFM, as well as fault models in other regions of the world.

The I^{2}GMM model may have applications to point processes beyond those arising in seismology. Space-time self-exciting point processes arise in the study of crime [37], [41], [42], conflict [43], and terrorism [44]–[47], as well as in social-network event dynamics, for example in social media [48]–[50]. In the case of crime, clusters arise naturally from the superposition of events committed by different offenders with different modi operandi. Similar clusters may arise from the operations of different terrorist groups within a geographic region. I^{2}GMM is a flexible model for capturing this type of clustering in the intensity of events of a point process.

2.3 The Analog Learning Framework

Besides AI^{2}GMM, we made another attempt on Non-Exhaustive Learning. Intuitively speaking if we can learn a feature space mapped by function f to improve the unsupervised clustering accuracy of known classes, the clustering accuracy of unknown classes in the feature space should also be improved. We proposed the analog learning framework (ALF) that aims to separate classes in the learned feature space without using the supervised information encoded in the class labels. Specifically, this framework has two stages. In the first stage the analog learning component attempts to learn the intrinsic metric that separates classes. In the second stage the validation component infers the assignments of samples in an unsupervised manner with metric information derived from the analog learning part. An evaluation score is used as an indicator of model fitness. With this approach the model is expected to draw an analogy between known and unknown classes by leveraging the past knowledge embodied in known classes.

2.3.1 Heuristic Parameter Learning for I^{2}GMM

Our first approach in the analog learning framework is built on top of I^{2}GMM. During our experiments with AI^{2}GMM we noticed that when there is a big discrepancy between the actual classes distributions and the model's assumption, performing restricted Gibbs sampling generates ill-defined components. Since the hyper-parameters are learned based on current cluster and components assignments, the learned hyperparameters further deteriorate the mismatch between true and predicted class distributions. When this happens, the performance of the restricted AI^{2}GMM with training data could be even worse than that of fully unsupervised version. Which is against our goal where we want to improve the performance of the model in discovering unknown classes through utilizing the supervised information provided by the training data.

Therefore, rather than performing restricted Gibbs sampling and learning the hyper-parameters based on the current assignments, we randomly sampled the hyper-parameters and perform fully unsupervised clustering using both training and testing datasets together. Then we evaluate the performance based on the assignment of the training samples using

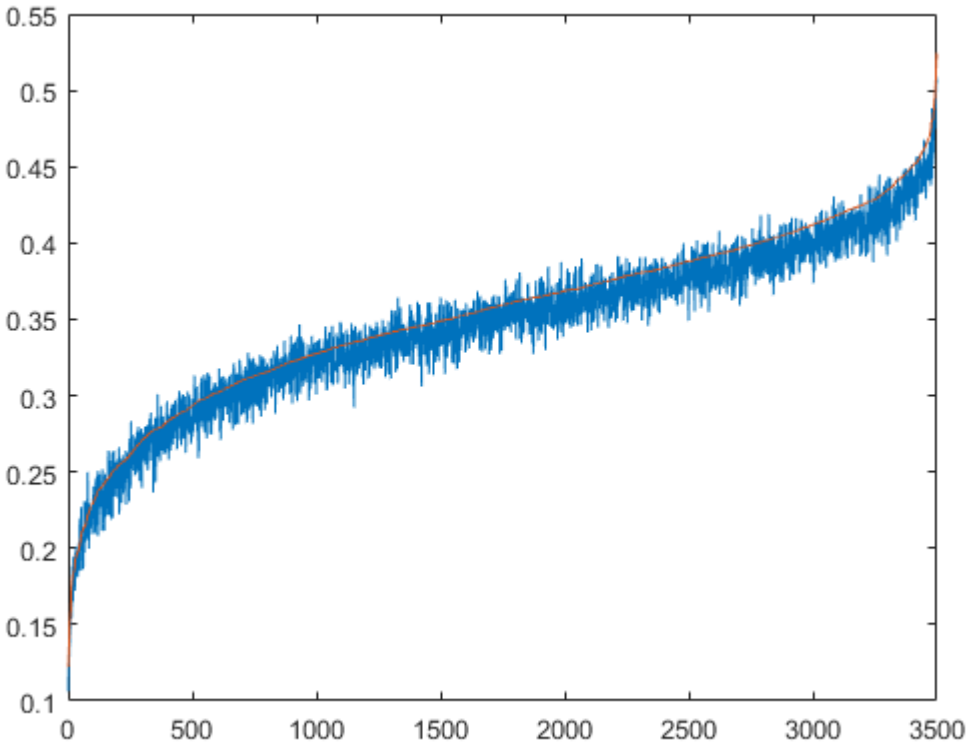


Figure 2.8. The comparison between training F1 and testing F1. Here we sort the 3500 runs of training scores and plot in sorted order (red line), then the testing scores of the same run is plotted at the same horizontal axis of the training score (blue line).

mean F1 score as the performance indicator of how good the selected hyper-parameters are. The hyper-parameters are then optimized by the Simulated Annealing (SA) algorithm [51] using the training score. In this method the SA hyper-parameter optimizing is the analog learning part and the clustering algorithm is the validation part of the analog learning framework. In order to see whether the assumption for the analog learning framework holds here, we performed experiment on the Statlog dataset from the UCI Machine Learning Repository [52], and plotted the comparison between training scores on training data and evaluation scores on test data in Figure 2.8. We can see from the figure that the testing score is trending higher with the training score with the exception of a small random perturbation.

Moreover, we can use the training score as an indicator of the goodness of fit between the model and the dataset. For datasets that do not suit the I^2GMM model, training score will still be low (higher is better) even after the SA algorithm converges but for datasets that fit the I^2GMM well, the training score will be high.

Although the heuristic method described above is effective and can eventually find the set of hyper-parameters that achieve the best performance, it is very computational costing since each time we evaluating the score of hyper-parameters suggested by the SA optimizer we need to run the I^2GMM from scratch using the updated hyper-parameters. Apart from the computational burden the applicability of this SA-based heuristics is limited to low dimensional datasets with well separated class distributions that I^2GMM could fit well. Performing NEL on complex datasets with structural information and high dimensional features requires learning a low dimensional feature representation. Rather than continue with this line of work, we decided to move forward to Non-Exhaustive Feature Learning through deep learning techniques.

3. NON-EXHAUSTIVE FEATURE LEARNING

In chapter 2 we discussed the theoretical work and applications of Non-Exhaustive Learning. Though considering NEL from a statistical aspect gives us theoretical support and helped with formulating the problem, it is limited to "plain" datasets whose statistical characteristics of each class is obvious and discriminating. This is usually not hold for many real word datasets such as natural images, texts, and speeches. Therefore, we put forwarded with the Non-Exhaustive Feature Learning (NEFL) task. Different from supervised or unsupervised representation learning which are extensively studied, in our NEFL work we focus on learning a statistically discriminating feature space that could separate well both known and unknown classes.

We first tackle this problem along the direction of generative models, and proposed the Variational Auto-encoding IGMM detailed in section 3.1. Then we changed our focus on the generalization ability of the feature learning method, and conducted our research under the metric learning direction. Section 3.2 gives an introduction of this approach. We then applied this method and proposed its two variants in two applications in section 3.3 and 3.4

3.1 Variational Auto-encoding IGMM

In order to generalize NEL to complex and structural datasets which usually appear with high dimensional and sparse features, we proposed the variational auto-encoding IGMM (VAIGMM) for non-exhaustive feature learning. VAIGMM is proposed based on the intuition that the IGMM prior can be made to more effectively fit a learned low dimensional feature space representation of the original features. Instead of picking an IGMM prior and fitting original data into this prior by brute force, we aim to jointly learn a nonlinearly projected representation of the original data and the IGMM prior that would best fit this data.

We resort to deep feature learning techniques since they are known for their flexibility and success in many domains. Feature learning can be either supervised or unsupervised. Supervised feature learning including Linear Discriminant Analysis (LDA) [53] and supervised neural networks learn a mapping f_θ to maximize the discrimination between different classes in the latent space based on labeled data. Since most of the supervised feature

learning methods are greedily maximizing the discrimination, the learned feature space is usually over-fitted for the known classes and generalize poorly on unknown classes. On the other hand, unsupervised feature learning techniques including Principal Component Analysis (PCA) [54], clustering based one hot encoding [55], auto-encoders [56] try to learn a feature space that could capture explanatory underlying factors that will be useful for high-level tasks such as classification [57]. The objective for unsupervised feature learning techniques is usually some similarity measure between the observations and the learned features, ensuring that the learned feature representations contains necessary information of the raw data with a small random perturbation. Due to the intrinsic objective, unsupervised feature learning techniques usually generalize well even on unknown classes.

Among all the techniques in unsupervised deep feature learning, variational auto encoder (VAE) [58] and generative adversarial networks (GAN) [59] are the most popular ones. While VAE imposes a prior for the latent layer and build a graphical model to make the latent layer distribution comply with the prior, GAN and its variants such as Adversarial Auto-encoders (AAE) [60] fit the latent distribution in a more straight forward way by imposing an additional network to distinguish between samples generated by the prior and encoded samples. We have tried both AAE and VAE for fitting an IGMM prior and finally choose VAE due to the better fit of data distributions it generate on the latent space. [61] proposed a model similar to ours for clustering by imposing a GMM on the latent space and achieved promising results on some benchmark datasets. The parametric mixture model such as GMM [61] uses requires the number of clusters to be defined beforehand. In our work we use IGMM as a non-parametric Bayesian prior and derive the corresponding evidence lower bound (ELBO) as the optimization objective for the feature learning neural network, and designed a training method to integrate labeled and unlabeled data in the NEL framework. Details are discussed in 3.1.1.

3.1.1 Method

In this section, we describe the generative model of VAIGMM along with the deriviation of its objectives. We also proposed a training method for integrated training of both labeled and unlabeled data in 3.1.1.

Generative Model for VAIGMM

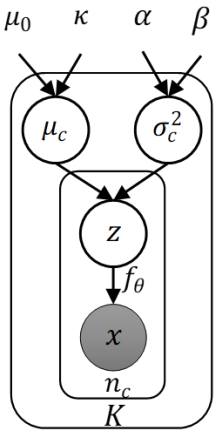


Figure 3.1. Directed graphical model for VAIGMM, where K is the current number of clusters found (could potentially be infinite), n_c is the number of data points in cluster c

The generative model for VAIGMM is below:

$$\begin{aligned} H &= N\left(\mu_c|\mu_0, \kappa^{-1}\sigma_c^2 I\right)\Gamma^{-1}(\sigma_c^2|\alpha,\beta) \\ G &= DP(\gamma H) \\ \theta_c = (\mu_c, \sigma_c^2) &\sim G \\ z &\sim N(\mu_c, \sigma_c^2 I) \\ (\mu_x, \sigma_x) &= f(z|\theta) \\ x &\sim N(\mu_x, \sigma_x^2 I) \end{aligned} \tag{3.1}$$

where H is the base distribution in the form of a Normal Inverse Gamma distribution with the expected cluster centers at μ_0. The scaling constant κ controls the separation among clusters.

The inverse Gamma distribution is defined by its shape parameter α and rate parameter β. A sample G is drawn from the Dirichlet process with base H and concentration γ; cluster center μ_c and scalar variance σ_c^2 are drawn from G, and latent space sample z is drawn from a Normal distribution with mean μ_c and covariance σ_c^2 times identity. Finally, latent space sample z is mapped by a function f parameterized by θ to generate the mean μ_x and variance σ_x in the observation space, and a data point is sampled by a Normal distribution centered at μ_x and covariance $\sigma_x^2 I$.

We choose the simplified version of IGMM with a spherical covariance for each cluster c in order to reduce the computational cost in high dimensions. Although this sacrifices the flexibility of the clustering model the neural network can still learn a feature space with well-separated and unimodal class distribution. This is the so called "thin" model and "heavy" feature learning. By doing this, we are training a feature learning network that could learn features useful for non-exhaustive learning.

According to the generative process in (3.1) and the graphical model illustrated in Figure 3.1, the joint probability $p(x, z, c)$ can be factorized as:

$$p(x, z, c) = p(x|z, c)p(z|c)p(c) = p(x|z)p(z|c)p(c) \tag{3.2}$$

since x is independent of c when conditioned on z. Based on the generative definition in (3.1), the probability terms in (3.2) are:

$$\begin{aligned} p(x|z) &= N(\mu_x, \sigma_x^2 I) \\ p(z|c) = p(z|\theta_c) &= N(\mu_c, \sigma_c^2 I) \\ p(c) &= DP(\gamma) \end{aligned} \tag{3.3}$$

Variational Lower Bound

We followed the work of [61] to derive the variational lower bound for our model. The training objective is to maximize the likelihood given the observations. According to the

generative process in 3.1.1, the log-likelihood of VAIGMM can be written as below by using Jensen's inequality:

$$\log p(x) = \log \int_z \int_c p(x, z, c) dc dz \geq E_{q(z,c|x)} \left[\log \frac{p(x, z, c)}{q(z, c|x)} \right] = L_{ELBO}(x) \tag{3.4}$$

where $\mathcal{L}_{ELBO}(x)$ is the evidence lower bound, $q(z, c|x)$ is the variational posterior to approximate the true posterior $p(z, c|x)$. We can factorize $q(z, c|x)$ as mean-field distribution and use a neural network g to model $q(z|x)$:

$$\begin{aligned} q(z, c|x) &= q(z|x)q(c|x) \\ q(z|x) &= N(z|\tilde{\mu}, \tilde{\sigma}^2 I) \\ [\tilde{\mu}, \log \tilde{\sigma}^2] &= g(x|\phi) \end{aligned} \tag{3.5}$$

and by approximating $q(c|x)$ with the posterior $p(c|z)$ we have:

$$q(c|x) = p(c|z) = p(\theta_c|z) = \frac{p(z|\theta_c)p(\theta_c)}{\int_{\theta_c} p(z|\theta_c)p(\theta_c)} = \frac{N(z|\mu_c, \sigma_c^2 I) N\left(\mu_0, \kappa^{-1}\sigma_c^2 I\right) \Gamma^{-1}(\sigma_c^2|\alpha, \beta)}{\int_{\theta_c} N(z|\mu_c, \sigma_c^2 I) N\left(\mu_0, \kappa^{-1}\sigma_c^2 I\right) \Gamma^{-1}(\sigma_c^2|\alpha, \beta)} \tag{3.6}$$

By integrating out the θ_c in the denominator, we have:

$$p(\theta_c|z) = \frac{\Gamma(\alpha + \frac{D}{2}) \beta^\alpha N(z|\mu_c, \sigma_c^2 I) N\left(\mu_0, \kappa^{-1}\sigma_c^2 I\right) \Gamma^{-1}(\sigma_c^2|\alpha, \beta)}{\left(\frac{\kappa}{1+\kappa}\right)^{\frac{D}{2}} \Gamma(\alpha) \left(\frac{\beta(1+\kappa)}{1+\kappa+\beta\kappa}\right)^{\alpha+\frac{D}{2}} N(z|\mu_0, I)} \tag{3.7}$$

Therefore, the $\mathcal{L}_{ELBO}(x)$ term can be rewritten as below according to (3.2) and (3.5):

$$\mathcal{L}_{ELBO}(x) = E_{q(z,c,\mu_c|x)}[\log p(x|z) + \log p(z|c) + \log p(c) - \log q(z|x) - \log q(c|x)] \tag{3.8}$$

After deriving each term based on their definitions in equations (3.3) and (3.5), and combining them together we have:

$$\mathcal{L}_{ELBO}(x) = \frac{1}{L}\sum_{l=1}^{L}\log N(x|\mu_x^{(l)}, {\sigma_x^{(l)}}^2 I) + \sum_{c=1}^{K+1} q(c|x)\left(\log\frac{\boldsymbol{\pi}_c N(\tilde{\mu}|\mu_c, \sigma_c^2 I)}{q(c|x)} - \frac{D\tilde{\sigma}^2}{2\sigma_c^2}\right) + \frac{D}{2}\log(2\boldsymbol{\pi}\tilde{\sigma}^2) + \frac{D}{2} \tag{3.9}$$

Note that we adopt the Stocastic Gradient Variational Bayes (SGVB) estimator [58] in (3.9), where L is the number of Monte Carlo samples. $\mu_x^{(l)}, {\sigma_x^{(l)}}^2 = f(z^{(l)}|\theta)$ with $z^{(l)}$ sampled from $q(z|x)$ in equation (3.5). By using the reparameterization trick, $z^{(l)}$ can be obtained with

$$z^{(l)} = \tilde{\mu} + \tilde{\sigma} \circ \epsilon^{(l)} \tag{3.10}$$

where $\epsilon \sim N(0, I)$ is the reparameterized random sample generated from a standard Normal distribution, $\circ$ indicate the element-wise multiplication. Different from the finite Gaussian mixtures, we included the term for generating new cluster in equation (3.9), which makes the sum term to be the number of existing clusters K plus 1. The estimated prior $\boldsymbol{\pi}_c$ is defined as below:

$$\boldsymbol{\pi}_c = \begin{cases} \frac{n_c}{N-1+\alpha}, & c \leq K \\ \frac{\alpha}{N-1+\alpha}, & c > K \end{cases} \tag{3.11}$$

Interleaving Training and Restricted Sampling for Non-exhaustive Learning

In Sections 3.1.1 and 3.1.1 we defined the generative model as a fully unsupervised model and derived the resultant optimization objective. The training procedure for non-exhaustive learning is described below:

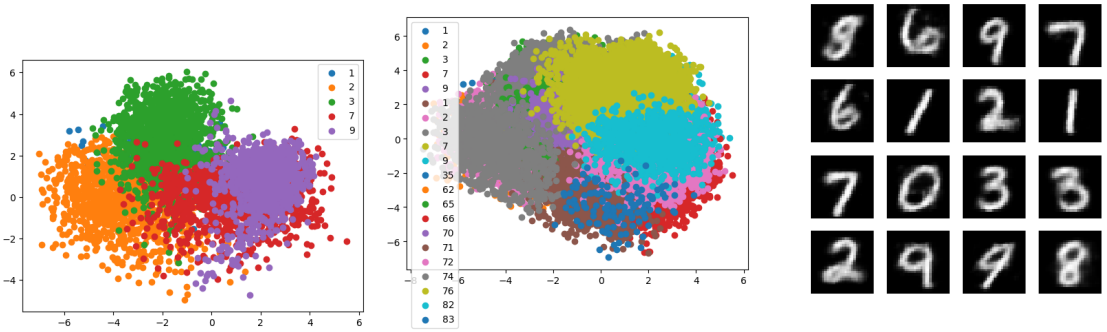
1. We pretrain a fully unsupervised stacked auto-encoder [14], initialize g_ϕ with the encoder and f_θ with the decoder. Then perform initial training on the feature space

encoded by g_ϕ using equation (3.5) with restricted IGMM using the same restrictive training strategy described in 2.1.2 on labeled data.

2. For each epoch, we perform one sweep of restricted Gibbs sampling similar to 2.1.2 for IGMM. Conditioned on the label, we maximize the ELBO in (3.9) by training the feature learning networks g_ϕ and f_θ. The second term in (3.9) for $c = K + 1$, i.e. the likelihood of assigning a sample to a new class can be estimated by replacing μ_c with μ_0, σ_c with the expectation of the inverse gamma $\frac{\beta}{\alpha-1}$.

3. Successively iterate training IGMM for cluster assignment and the variational networks for feature representation learning until convergence.

3.1.2 Experiments



(a) Plot of first 2 dimension on the feature space for encoded training data.

(b) Plot of first 2 dimension on the feature space for encoded testing data.

(c) Images of decoded cluster centers.

Figure 3.2. Illustration of MNIST after training VAIGMM for 200 epochs on 5 of the digits.

Due to the inappropriate inference of previous version, I don't have the experiments results for current version yet. The previous version and the VAIGMM described here differ in the generative model, where in the previous version the cluster covariance are all assumed to be identity; and the previous version use $p(x, z, c, \mu_c)$ as the joint probability in (3.2) and making too many approximations. Figure 3.2 showed an illustration of running previous version of VAIGMM on MNIST with feature space dimensions set to 10. We can see from

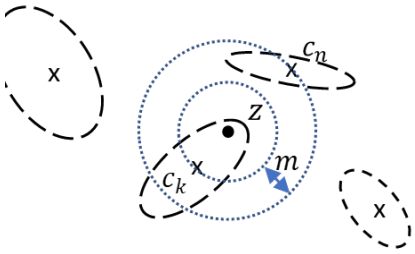


Figure 3.3. The illustration for centroid margin loss. Where "x" mark indicate the class center, "•" indicate the sample point, m is the margin.

the figure that though the training data is well separated, most of the testing classes are still mixed together. From the generated digits by decoding cluster centers, we can observe that only few of new digits are discovered, and some of the classes are split into different clusters by their different shape variation. Based on previous results, relying on restricted Gibbs sampling to learn the supervised information from the labeled data seems not generalize well on unknown classes. Which means, we got high accuracy on known classes but doesn't perform well on discovering new classes. But this could be due to the inappropriate inference and the choice of $q(c|x)$. Note that we approximate $q(c|x)$ with $p(c|z)$ in (3.6). Rather than use the posterior distribution we can also use the posterior predictive $p(z^*|c)$, which might be a better choice than the posterior.

3.2 The Centroid Margin Loss

Centroid Margin Loss (CML) is inspired by the triplet loss [62], which is a metric learning technique that avoid using explicit class labels but still encourage samples coming from same classes to be close to each other and far away from other classes. Though triplet loss is known to generalize well on unknown classes for transfer learning [63], it is also known for its less desirable convergence characteristics due to the large number of potential triplet (n^3) that can be tried [63]. We proposed the stochastic centroid margin loss to deal with this problem. For each batch with N data, we sample N_{known} labeled data points from each of the K classes, and $N_{unknown}$ data points from unlabeled data. For a labeled data point with feature z in class k, we evaluate the centroid c_{i} of each class in the feature space, and use the centroid

of its own class c_k as positive sample. Then we use the centroid c_n closest to x among all centroids except c_k as the negative sample. Then the centroid margin loss is calculated as:

$$L(z, c_k, c_n) = max\left\{d\left(z, c_k\right) - d\left(z, c_n\right) + margin, 0\right\} \tag{3.12}$$

Note that this equation is exactly the same as triplet loss but with positive samples replaced by the centroid of its class of origin, and the negative samples replaced by the centroid of the closest class. We have illustrated the centroid margin loss in Figure 3.3. By replacing the positive random sample from the same class of the anchor point by its class centroid, and replacing the negative sample by its closest negative centroid, we highly improved the effectiveness of the triplet loss. When performing random sampling most of the negative sample distances associated with the anchor point will be larger than the margin associated with the positive sample distance resulting in too many trivial triplet loss of 0s. After replacing the random samples with centroid, we reduced the number of potential triplet from n^3 to n, which is the same as the number of data points and overcomes the convergence problem of triplet loss.

We compared the triplet loss and CML on MNIST by using them as an optimization objective on the feature space of a stacked auto-encoder. The stacked auto-encoder is first pretrained with all data in a fully unsupervised manner, and then fine tuned by triplet loss or CML on the training data. Then we perform K-means clustering algorithm on the encoded features to generate the label, the results are showed in Table 3.1. In this experiment, training data are sub-sampled from the original MNIST train split with 80% of data points from 5 of the classes, validation data is the rest of the MNIST training data, and the testing data contains all data points in the MNIST testing split. We can see from the comparison that both the triplet loss and centroid margin loss generalize well on unknown classes but centroid margin loss achieves much better results than triplet loss.

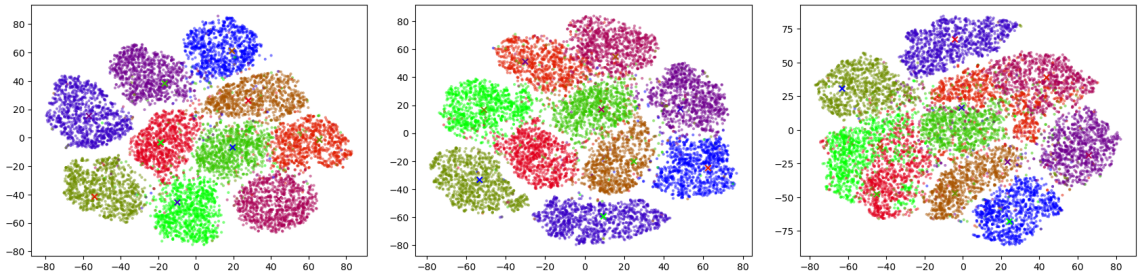
The main challenge to perform feature learning using centroid margin loss for NEL is how to utilize the unlabeled data. In the experiments above both the centroid margin loss and the triplet loss are optimized with only labeled data. The unlabeled data is only used in the auto-encoder. Recent studies suggest that when learning the feature space keeping

Table 3.1. Comparison of mean F1 scores of Centroid Margin Loss and Triplet Loss on MNIST

	Train F1	Validation F1	Testing F1
Centroid Margin Loss	0.95	0.89	0.90
Triplet Loss	0.88	0.75	0.75
cross entropy loss	0.96	0.62	0.62

the sample x and its random transformation in the same class will significantly improve the performance [9], [16]. Herein we define this kind of transformation to be "class invariant", as these transformations are not supposed to affect the class semantics of the data. Moreover, the class invariant transformation provides us a way to incorporate task-specific domain knowledge into the model. We proposed the unsupervised centroid margin loss based on the class invariant transformations. For each unlabeled data, we generate l class invariant transformations, and consider the centroid of these l data points as the positive centroid for the current unlabed data point. We then choose two of the closest centroids from labeled data, generate m samples by blending transformations of the two centroids, and treat the centroid of these m samples as the negative centroid. We have illustrated the feature space learned by CML with unlabeled data, CML only on labeled data, and the triplet loss learned on labeled data using t-SNE in Figure 3.4. We can see from the illustration that centroid margin loss with unlabeled data get the most clear class boundaries, and triplet loss is the worst among them.

In order to see how our proposed model compares with state-of-art, we performed experiments on Omniglot dataset and compared with Deep Transfer Clustering (DTC) [9]. We adopted the same network architecture as them and loaded their pretrained model, along with the same labeled and unlabeled data split. Due to the limitation of batch size constraint by GPU memory size, we can't load all training classes all at once, since centroid margin loss requires sampling N_{known} points from each training classes in every batch. Therefore we trained the target alphabet against every training alphabet for 10 epochs and perform KMeans clustering to generate the label, then align all the labels and take the mode as the



(a) t-SNE illustration for MNIST testing data on the feature space learned by optimizing centroid margin loss on both the labeled and unlabeled data.

(b) t-SNE illustration for MNIST testing data on the feature space learned by optimizing centroid margin loss only on labeled data.

(c) t-SNE illustration for MNIST testing data on the feature space learned by optimizing triplet loss on labeled data.

Figure 3.4. Comparison of t-SNE illustration for MNIST testing data on the feature space. Where we use different color to illustrate different classes.

final label for the target alphabet. This process is repeated for 10 epochs and the accuracy for the last epoch is reported. The comparison results is showed in Table 3.2.

Table 3.2. Comparison of accuracy on Omniglot

	DTC-π	DTC-TE/TEP	Centroid Margin Loss
ACC	89.0 %	87.8 %	88.1 %

Though simple and trained only for 10 epochs, centroid margin loss still achieves comparable performance with the DTC on Omniglot. Moreover, in this experiment the set of classes in the training and testing data sets are mutually exclusive, which DTC is designed for. DTC split the learning of labeled data and unlabeled data into pretraining and fine tuning stages, whereas there is no information about unlabeled data in the pretraining stage and also no labeled data is used in the fine tuning stage. We don't know how DTC will perform when labeled data is included as in the non-exhaustive learning scenario. DTC is based on DEC [64], which optimize the likelihood of the clustering on the feature learning network. In our case, we can use the unsupervised version of VAIGMM introduced in section 3.1 rather than KMeans clustering for the validation part of the analog learning framework. By jointly maximizing the ELBO of VAIGMM and the triplet margin loss, the model may

be able to learn a feature space that not only the classes are well separate but also cope with the IGMM prior. This could be a line of future research direction.

3.3 Applications to Segmentation of Sub-cell Structures in Multiplex Stimulated Raman Scattering Images by Open World Feature Learning and Nonparametric Bayesian Clustering

In this section, we discussed the application of Centroid Margin Loss on a hyper-spectral image datasets. We proposed a variant of CML that is robustive to outliers or tagging errors in order to tackle with the challenges appeared in the application.

3.3.1 Introduction

Semantic segmentation is a popular research area in machine learning with many applications in segmentation of objects in street scene photos [65], cells in microscopy images, landcover types in hyperspectral images, just to name a few. Currently successful semantic segmentation algorithms use fully convolutional neural (FCN) networks such as UNet [66], or HRNet [67], and their performance heavily rely on contextual learning. Unlike semantic segmentation in RGB or Gray scale images, in this project we deal with the semantic segmentation of multiplex stimulated Raman scattering (SRS) images to identify static (nucleus, endoplasmic reticulum) as well as dynamic (lipid droplets) sub-cell structures.

SRS imaging cytometry is a label-free single-cell analysis platform with chemical specificity and high-throughput capabilities [68]. This platform can generate hyperspectral images with dozens of channels that reflect the chemical components inside cells with high spatial and temporal resolution. The images can be analyzed to delineate a map of chemical activities inside single living cells paving the way for discovery of important molecular events in cell cycle.

In this paper we try to find solutions to the following key challenges of semantic segmentation in SRS images. (1) How can we learn a feature space that can preserve separability among classes representing known structures but at the same time allow for the discovery of subclasses representing interesting sub-cell structures that emerge with unknown spectral patterns? (2) How do we utilize spatial information in order to reduce noise for a more

robust spectral segmentation? (3) How do we perform segmentation without knowing the number of spectral patterns in each image beforehand. In order to deal with the first two challenges, we propose a 3D CNN Auto-encoder model [69] trained with a newly proposed contrastive loss function to learn a new feature space specific enough to distinguish classes representing known static structures but general enough to facilitate discovery of new classes representing dynamic structures. In order to deal with the third challenge we use a doubly non-parametric hierarchical Bayesian model [12], [70] to cluster pixel embeddings in the new feature space to identify spectral patterns with potentially significant biological and chemical implications.

3.3.2 Weakly Supervised Label Generation

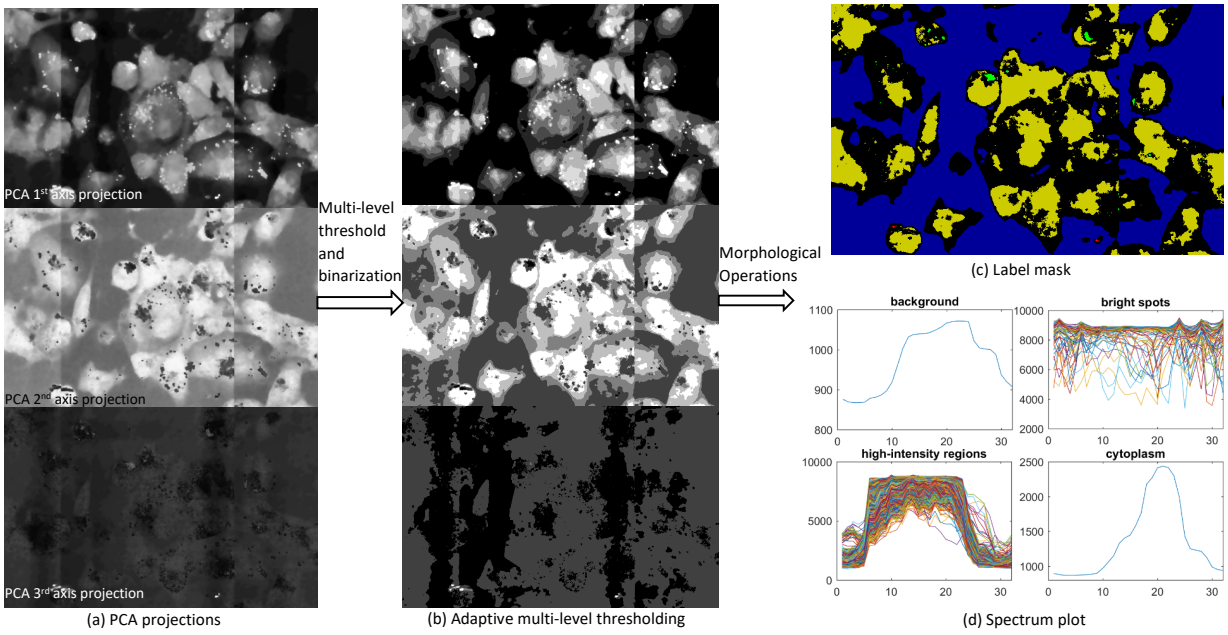


Figure 3.5. Illustration of the label generation pipeline. Where in (c) black is unknown area; blue is background, yellow is cytoplasm, green indicate high-intensity regions, and red is bright spots; (d) for better illustration we plot the mean of background and cytoplasm, and plot the bright spots and high-intensity regions classes individually.

One trivial approach to segment SRS images involves thresholding gray level images obtained by the first few principal components. As different principal components capture the

maximum variance along a different set of orthogonal directions they can be used to segment different compartments of the image provided that there is enough intensity difference among these compartments. For example the first principal component can segment cells from the background and the second principal component when used together with the first one can identify high intensity regions inside cells as long as their gray-level images are binarized at suitably adjusted thresholds. Although such an approach would not be robust to segmentation of less common and subtle sub-cell structures due to variations in overall image characteristics, a coarse-grained segmentation map that identifies background, cytoplasm, high-intensity regions, and artifacts, usually characterized by very bright spots, can still be obtained by PCA analysis.

We propose a coarse label generation process based on the adaptive multi-level thresholding of gray-level images as illustrated in Figure 3.5. (a) In the first stage we pre-process the pixel-scale spectral data by a median filter, and then project this data into a 3-dimensional PCA space defined by the first three principal components; And for each PCA projection of the data, we pre-processed with the Adaptive Histogram Equalization (CLAHE) [71] followed with a wiener filter [72] to remove the effect of noises and uneven exposure; (b) In the second stage we calculate multi-level thresholds using Otsu's method [73] and tune the number of bins for each PCA axis; (c) In the third stage we obtain the segmentation masks for each class by binarizing gray-level PCA images using thresholds from the previous stage, and further refine the results by morphological operations such as morphological openings and closing, and finally combine these binary images to segment four main compartments: background, cytoplasm, high-intensity regions, and bright spots.

In this study we show that these coarse labels can be useful for weakly supervising the training process to discover subtle spectral patterns representing dynamic structures when a suitably chosen network architecture and loss function can be selected.

3.3.3 Method

Given the coarse-grained labels generated in Section 3.3.2, we propose a 3D convolutional auto-encoder architecture to perform fine-grained segmentation. The high-level block dia-

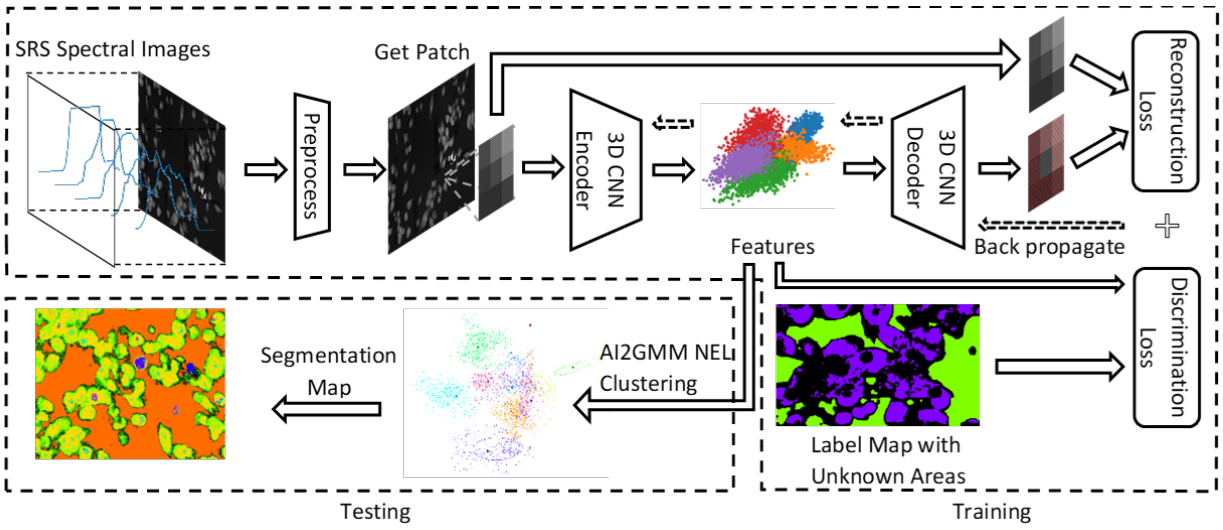


Figure 3.6. Illustration of the training and testing workflow.

gram of the training and testing stages are illustrated in Figure 3.6. (1) Given a set of SRS images with each pixel representing a signal of C channels, we perform preprocessing that involves signal-wise normalization and whole image re-scaling and normalization. (2) Using preprocessed images we extract p by p patches centered at pixel locations (i, j). Note that p needs to be an odd number in order to avoid asymmetric sampling. These patches are used to train a CNN-AE described in Section 3.3.3. (3) We train the network by simultaneously optimizing the reconstruction loss and the discrimination loss as described in Section 3.3.3. For patches for which the labels are unknown, we only optimize the reconstruction loss. (4) After the network is trained, patches are embedded into the feature space generated by the CNN-AE to obtain pixel embeddings. These pixel-scale embedding vectors are clustered by a doubly nonparametric Bayesian model as described in Section 2.1.2. During clustering pixels from background class are not used. Finally, segmentation maps are generated based on the cluster indicator variables for each pixel.

We describe the details of the 3D CNN-AE in Section 3.3.3 and the proposed discrimination loss in Section 3.3.3. We also included an ablation study on each part of the model component in subsection 3.3.3.

3D Convolutional Autoencoder for Weakly Supervised Hyperspectral Image Segmentation

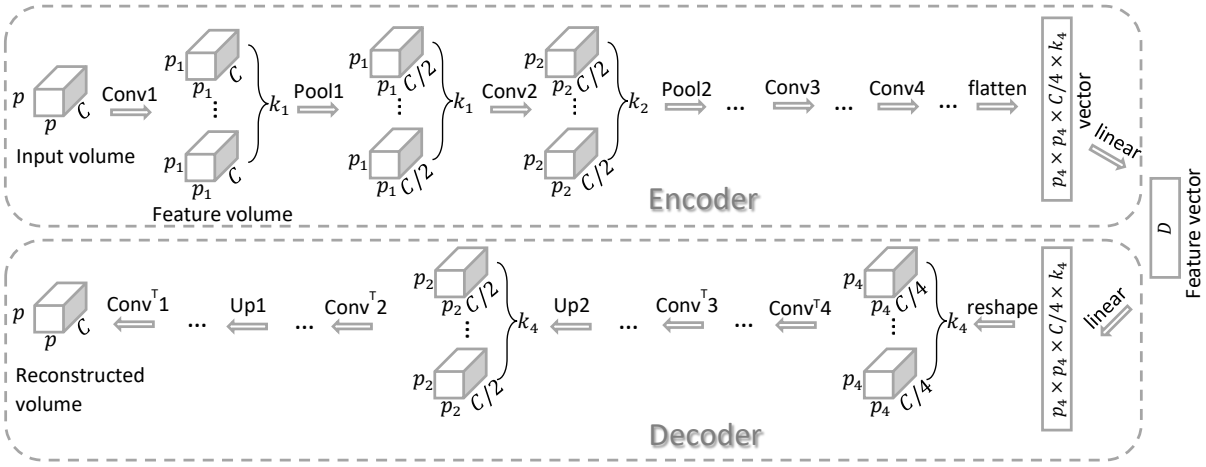


Figure 3.7. Illustration of the 3D CNN-AE network architecture.

3D Convolutional Neural Net (CNN) uses spectral and spatial information together and can significantly reduce image noise to generate spatially more coherent segmentation maps. However, when the goal is to discover dynamic sub-cell structures there is always the risk of eliminating subtle spectral patterns representing these structures as noise unless conservative training schemes are adopted. To obtain a more accurate segmentation map of both dynamic and static cell structures we use a 3D CNN-AE architecture. The backbone network generates feature embeddings for each pixel data. The decoder reverses the operation of the backbone network to reconstruct the input patch given the feature embedding. The network end-to-end is trained to optimize a loss function combining discriminant and reconstruction losses.

For the backbone network there are many options. For example we can use the most basic Multi Layer Perceptron (MLP), or a 1D convolutional neural network implemented in the spectral space, or a 2D convolutional neural network implemented in the spatial space. However, none of these models consider spatial and spectral information together. Multiplex SRS generates hyperspectral images and our work can benefit from a diverse set of CNN architectures successfully applied to remote sensing data [74]. Upon preliminary evaluation of different architectures we choose the model in [69] for its superior classification performance

as our base model. This architecture take full advantage of the spatio-spectral information and has lower computational costs.

We added a decoder part in order to deal with challenges (1) and (3). Since there exist unknown areas which is highly possible to contain new classes, training relies only on supervised information would tend to over-fitting. Moreover, since PCA gives us many useful information on classification as showed in section 3.3.2, we believe by setting a bottleneck at the coding space and learning a reconstruction loss will also benefit the performance. And by the combination of reconstruction loss and discrimination loss and training across images from different time shift, we are expecting that by feature reduction we can filter out the information that related to intro-class heterogeneity.

We build the decoder as the reverse architecture of the backbone network as illustrated in Figure 3.7. Where in the figure Convi is the ith convolution layer composed of k_{i} $3\times3\times3$ 3D convolution filters followed with a Relu activation layer; Pooli is the ith down-sampling layer composed of same number of 3D convolution filters as previous layer with stride 2 across channels (the spatial stride is still 1); $Conv^T$i is the reverse convolution layer composed of $k_{\mathrm{i}-1}$ $3 \times 3 \times 3$ 3D transposed convolution filters and a Relu activation layer similar to Convi, but this time we put the Relu layer before the 3D filters; Upi is the corresponding up-sampling layer using transposed convolution filters with stride 2 across channels; flatten and reshape are the pair of operations that flatten a 4D feature volume into a vector and reshape a vector back to the feature volume; linear is a simple fully connected layer map a x dimensional feature vector to a y dimensional one; p is the patch size, p_{i} is the spatial size after convolution layer i, C is the number of channels. Note that, since the number of channels is not always an even number when performing pooling, the last channel will be omitted during convolution. Even though we can pad both side of the feature volume before pooling to avoid lose of information, but this will still result in miss-match of the number of channels in the reconstruction phase to its corresponding down sampling one. To solve this problem, we adopted the output padding to restore the original shape.

When training the network, we optimize the sum of reconstruction loss and discrimination loss. Note that for reconstruction loss, in order to be consistent with discrimination loss and to avoid marginal effect (which could be very severe since we use a very small patch size

and padded the features), we only reconstructed the center pixel. Though the output of the decoder is of the same shape as input, we calculate reconstruction loss only between the center pixel and its reconstruction. We use Normalized Spectral Similarity Score (NS3) [75] as the reconstruction loss and re-scaled the data in order to make both of the loss at same magnitude. For cross entropy loss we added a fully connected layer on the feature space in order to generate desired feature dimensions.

Centroid Margin Loss

We proposed the Centroid Margin Loss (CML) in section 3.2 for learning a feature space that has good discrimination on both the training classed and could generalize well on unknown classes. The generalization ability is meanly due to the non-greedy optimization characteristics of CML. Since the samples will not contribute anymore in the loss once satisfied the discrimination criterion, the learned feature space is not tend to over-fitting to the training data. And as a contrastive learning objective, CML doesn't make any assumption on the class distributions or the feature space, which gives high flexibility for it to kept most of the original data relationship. We restated the formulation of CML as below:

For each batch of N data samples composed by K classes, we evaluate the centroid c_{i} of each class i in the feature space. Then for a labeled data point with feature z in class k, we use the centroid of its own class c_k as positive centroid, and note the centroid which is closest to x among all other centroids except c_k as the negative centroid c_n. Then the centroid margin loss is calculated as:

$$L(z, c_k, c_n) = max\left\{d\left(z, c_k\right) - d\left(z, c_n\right) + m, 0\right\} \tag{3.13}$$

Where m denotes the margin. We illustrated CML in figure 3.3. From the illustration we can see that CML is trying to find a feature space that each sample is at least m far away from its closes negative centroid then its own centroid. Which means, their should be a at least m size margin between the boundary of each class.

Due to the existence of outliers and incorrect labeling in the SRS spectral images, we proposed a variants of CML, which we called kernel CML (K-CML). We observed the phe-

nomenon of feature space collapse when training with the basic CML after a few epoch, especially when the data is not re-scaled. In the case of feature space collapse, the CML loss increases rather than reduce after a few epochs, and finally all losses becomes m. When checking the features for data samples, we noticed that they all mapped to a same location. Therefore, $d(z, c_k) - d(z, c_n)$ will always be 0, which makes the final loss to be m. Our inspection on the training process indicate that this is due to the large gradient generated by high error of the outliers. So we proposed CML with sigmoid kernel in order to smooth the loss at outliers:

$$L(z, c_k, c_n) = max\left\{\phi\left(d(z, c_k) - d(z, c_n)\right) + m), 0\right\} \tag{3.14}$$

where $\phi(x)$ is the kernel which could be of any form. For our scenario, we choose $\phi(x) = sigmoid(x) - 0.5$ in order to make the range of $\phi(x)$ between -0.5 and 0.5 and centered at 0. By doing so, no matter how large is the outlier in the feature space, it could contribute at most 0.5 with a much smoother gradient in the loss.

List of Compared Models

We did an ablation study for different aspects of our proposed model. Below is a list of the descriptions for all compared models.

1. **3D CAE with KCML loss and AI2GMM**. This is our proposed model. For this model, we use the 3D CAE described in section 3.3.3 along with the sigmoid kernel CML in 3.14 to generate feature. And we set the coding space dimensions to be 10D. We then obtained the label by running a constrained AI2GMM on the features as described in 3.3.3.

2. **3D CAE with CEL and AI2GMM**. We evaluate the generalization ability of KCML loss by compare it with standard cross-entropy loss (CEL). For this model, everything else is the same with our proposed model except that we use cross-entropy loss instead of KCML loss. Note that since the number of dimensions of the feature space is usually not equal to the number of classes, and cross-entropy prediction is known not good for

using as feature, we added a linear layer on top of the bottleneck layer of the CAE for CEL.

3. **3D CAE with AI2GMM**. In this model we removed the discrimination loss part and only optimize the reconstruction loss when training the CAE. Everything else is the same as proposed model.

4. **3D CNN with KCML loss and AI2GMM**. To evaluate the effectiveness of the auto-encoder framework, we removed the decoder part and the reconstruction loss. Which means, we train a network of the same architecture of [69] except for the last layer we replace the cross-entropy loss with KCML loss. And after training we take the feature as the last layer output and run a constrained AI2GMM on the feature the same as our proposed model.

5. **3D CAE with KCML loss and KMeans**. In this configuration we compared the performance of AI2GMM and KMeans. We use the same feature as our proposed model, but instead of running a AI2GMM, we run a Kmeans with various K values and use the one get the best results, which is 5 in our experiment.

6. **PCA feature with AI2GMM**. Rather than using features from CAE, we tested with the PCA features the same dimension with the CAE feature. And then run an AI2GMM model the same as in our proposed model.

7. **PCA feature with KMeans**. Same as above except that we replace AI2GMM with KMeans.

8. **Original Signal with AI2GMM**. We also tested how the results will be if we run an AI2GMM model on the original signals with all 32 channels.

For all these models we runs on the same training testing split and use the same pre-processing method. For the neural network part we trained with 10 epochs with stochastic gradient decent using Adam optimizer. The results are shown in section 3.3.4.

3.3.4 Experiments

We did the experiment on SRS images collected on different starvation time. In order to evaluate the generalization ability of the compared methods, we select two images from control group and one image from 6h as training images, and select one from each of the control (0h), 6h, 12h, 24h, and 48 hours starvation time as testing images. We further split the training data by 90% pixels in training and 10% pixels in validation when training the neuron network. We generated the weakly supervised label as described in section 3.3.2 composed of 4 coarse classes (background, cytoplasm, other components, and artifacts) and unknown areas. And use these labels to train our model. Due to the highly unbalance of the 4 classes (where the size of the smallest class is around 0.007% of the greatest one), we balanced the class size by a customized sampler when training the neuron network. We then run the experiments with models listed in section 3.3.3 and generate labels using AI2GMM or Kmeans for the testing images. Since it take too long to run AI2GMM on the whole image (which usually has millions of pixels), we did a sampling by VCA [76] to reduce the sample size, and then perform inference on the sampled data. For fair comparison, we did the same sampling when perform KMeans. In order to obtain the segmentation map for the whole image, we did a single Gibbs sampling sweep for AI2GMM (for KMeans we assign the rest of the pixels to their nearest centers).

We did a quantitative evaluation of the clustering results on the VCA samples. In order to match the labels between the ground truth classes and the clusters, we first did an alignment by Hungarian algorithm. And then calculate the macro F_1 3.15 and micro F_1 3.16 evaluation same as in classification: 3.16:

$$F_{1_{macro}} = \frac{1}{K} \sum_k \left(\frac{2TP_k}{FP_k + FN_k + 2TP_k} \right) \tag{3.15}$$

$$F_{1_{micro}} = \frac{2TP}{FP + FN + 2TP} \tag{3.16}$$

Where TP, FP, FN means the number of true positive, false positive, and false negative samples for the whole data set; TP_k, FP_k, FN_k denotes the true positive, false positive, and

false negative samples for class k; and K is the total number of ground truth classes. We can see from the equation that micro F_1 is the global score evaluate the performance with regard to the total percentage of pixels; while macro F_1 is the average F_1 score for each class and treated each class equally no matter how many samples falls in them. And the macro F_1 usually has more practical meanings, since we cared more about rare classes. The results are shown in table 3.3.

Table 3.3. Macro and Micro F1 scores for the compared models.

	ctrl	6h	12h	24h	48h	avg
3D CAE + KCML + AI2GMM	**0.89 0.92**	**0.92 0.84**	**0.91 0.82**	**0.92 0.89**	**0.96 0.94**	**0.92 0.88**
3D CAE + CEL + AI2GMM	0.84 0.9	0.9 0.77	0.88 0.77	0.81 0.71	0.78 0.89	0.84 0.81
3D CAE + AI2GMM	0.72 0.76	**0.92 0.84**	0.85 0.76	0.78 0.77	0.85 0.85	0.82 0.80
3D CNN + KCML + AI2GMM	0.58 0.51	0.79 0.67	0.81 0.58	0.79 0.68	0.61 0.7	0.72 0.63
3D CAE + KCML + KMeans	0.46 0.71	0.41 0.81	0.36 0.73	0.39 0.58	0.36 0.64	0.40 0.69
PCA + AI2GMM	0.76 0.64	0.85 0.83	0.78 0.6	0.8 0.66	0.74 0.7	0.79 0.69
PCA + KMeans	0.21 0.41	0.19 0.35	0.23 0.37	0.3 0.41	0.24 0.39	0.23 0.39
Original signal + AI2GMM	0.59 0.41	0.61 0.49	0.67 0.54	0.65 0.62	0.79 0.66	0.66 0.54

We can see from the quantitative results that our proposed method performed best among all the 8 methods. Also, AI2GMM gave a large performance drump over KMeans especially in terms of macro F_1. From the score of 3D CAE (0.92 0.88) vs 3D CNN (0.72 0.63), we can conclude that by adding a decoder part and introducing the reconstruction loss helped a lots. Moreover, we can see that with 3D CAE alone we obtained good results (0.82 0.80), and adding the discrimination loss by cross-entropy doesn't help a lot (0.84 0.81). But we got the performance drump by replacing CEL with KCML loss (0.92 0.88).

In additional to the quantitative evaluation by F1 score, which doesn't take into accounts of unknown areas, we also did a qualitative evaluation by visual inspection on the generated segmentation map and spectral. We first aligned the clusters with the weakly supervised classes using Hungarian algorithm, and label the rest of the clusters increasingly. Since the number of generated clusters are different for each method, it is not possible for us to use same coloring schemes for all method. In order to get better visualization, we use same colors for the clusters aligned to known classes. For the segmentation map we cropped the original segmentation map to 2400×1800 pixels for better illustration since their sizes varies

between images. To illustrate the spectral, we plot the averaged 32 channels spectral of the original image for each generated cluster. And we also use the same color scheme for the spectral plot and its corresponding segmentation map.

We compared the segmentation map and the averaged cluster spectral over the most basic method to the most advance model in figure 3.8 and figure 3.9. From the segmentation map, we can see that with discrimination loss, we get cleaner background, and get better results for the cytoplasm class (showed in cyan, please refer to class 3 in the legend of spectral plot). Moreover, the size of the cluster mapped to unknown area is smaller than other methods, from which we can think that our proposed method did better job in splitting the unknown area to their real class. For the segmentation map both our proposed method and PCA+AI2GMM did a good job on identifying cell boundary and their different compartment, while PCA+KMeans failed to do so. The results of PCA is greatly affected by the stripped noises. As for the comparison of the averaged spectral over generated clusters, we can see that both of our proposed method and PCA+AI2GMM did well on identifying the major compartments. But our proposed method generate fewer overlapping clusters. While we can see that PCA+KMeans failed to capture the patterns.

In additional to the comparison between different methods, we also did an analysis of the segmentation map and the corresponding spectral of our proposed method in figure 3.10 and figure 3.11. We can see that our proposed method successfully discovered sub-classes of the coarse class split. Note that since most of the sub-classes found are in the unknown area, we wouldn't lose the score for discovering sub-classes when perform quantitative analysis. From the joint analysis of the segmentation map and the spectral plot, we can see that even though they have similar shape, the magnitude of the cytoplasm increasing when get closer to the cell center; the patterns of sub-classes belongs to others class varies not only in magnitude but also in their shape, and they usually diverse when the environment of the cell changed (the pattern of "others" class in the center of a large cell usually differs with those in small ones and on the boundary). We can conclude from the joint analysis that our proposed method successfully handled challenge (2).

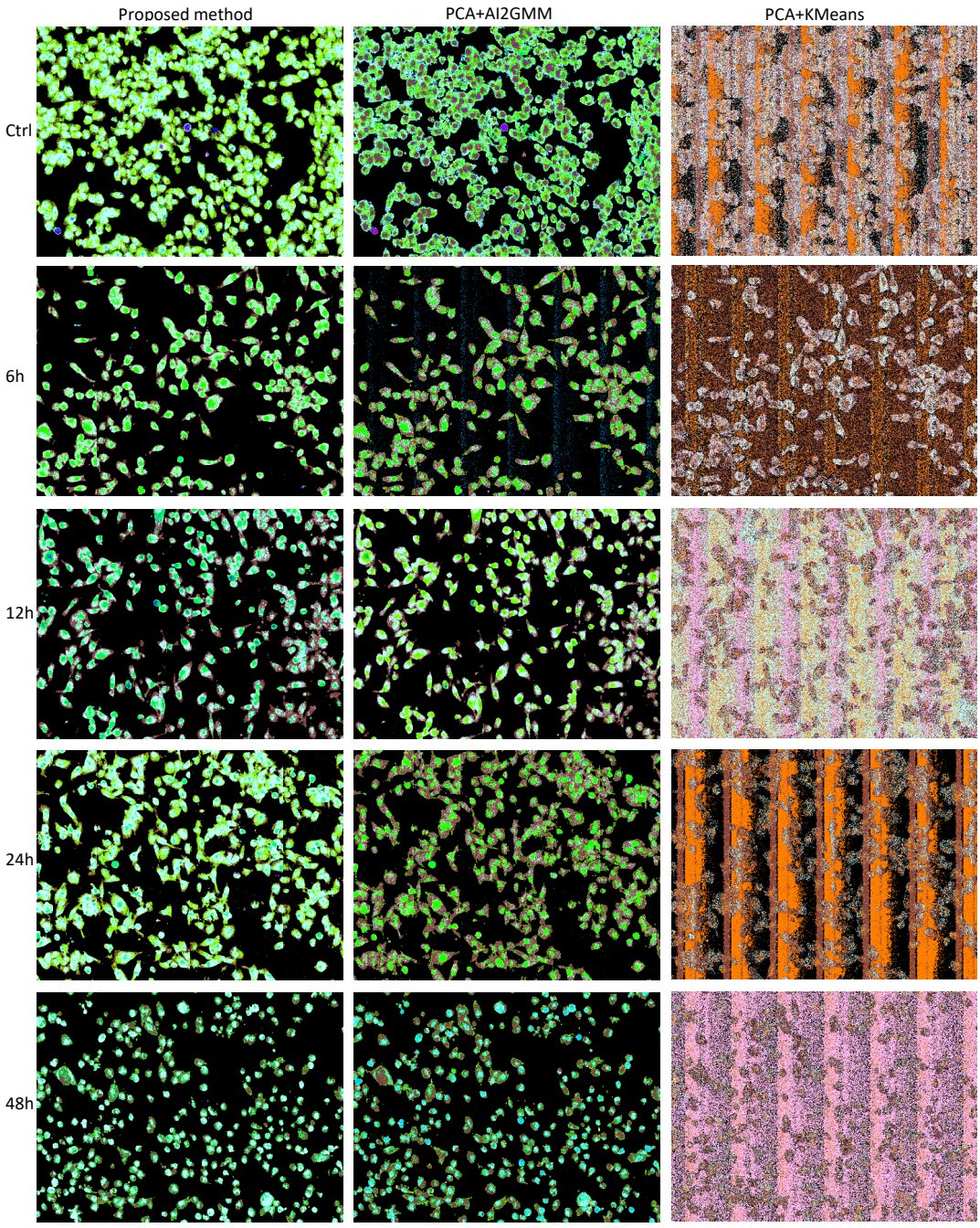


Figure 3.8. Comparison of the segmentation map for our proposed method and 2 other baselines.

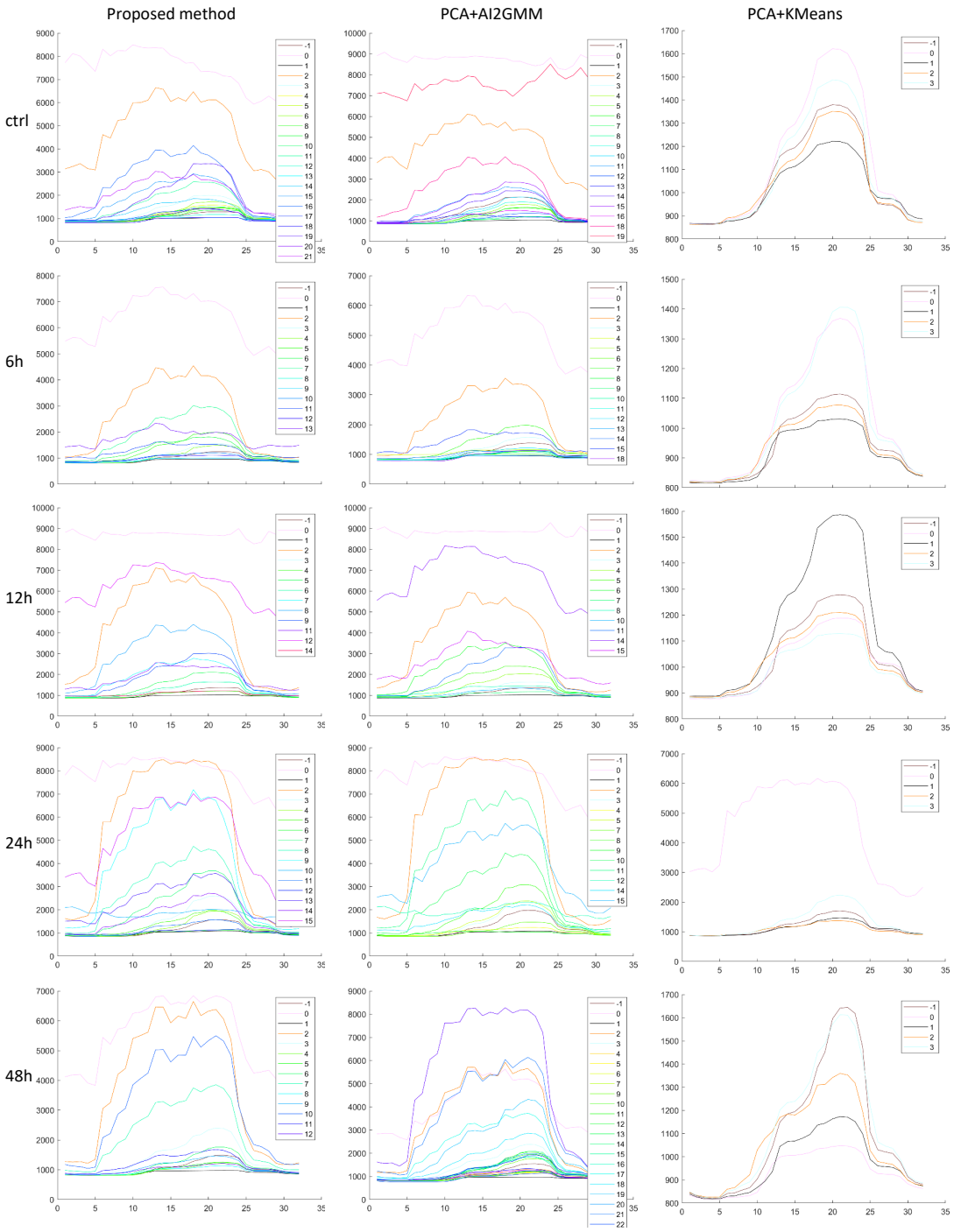


Figure 3.9. Comparison of the averaged spectral for each generated cluster of our proposed method and 2 other baselines. We skipped small clusters with less than 50 samples which might be generated due to random noises. The legend is the cluster labels, where -1 is the cluster aligned to unknown area, 0 is artifact, 1 is background, 2 is others, 3 is cytoplasm, and the rest are newly generated clusters.

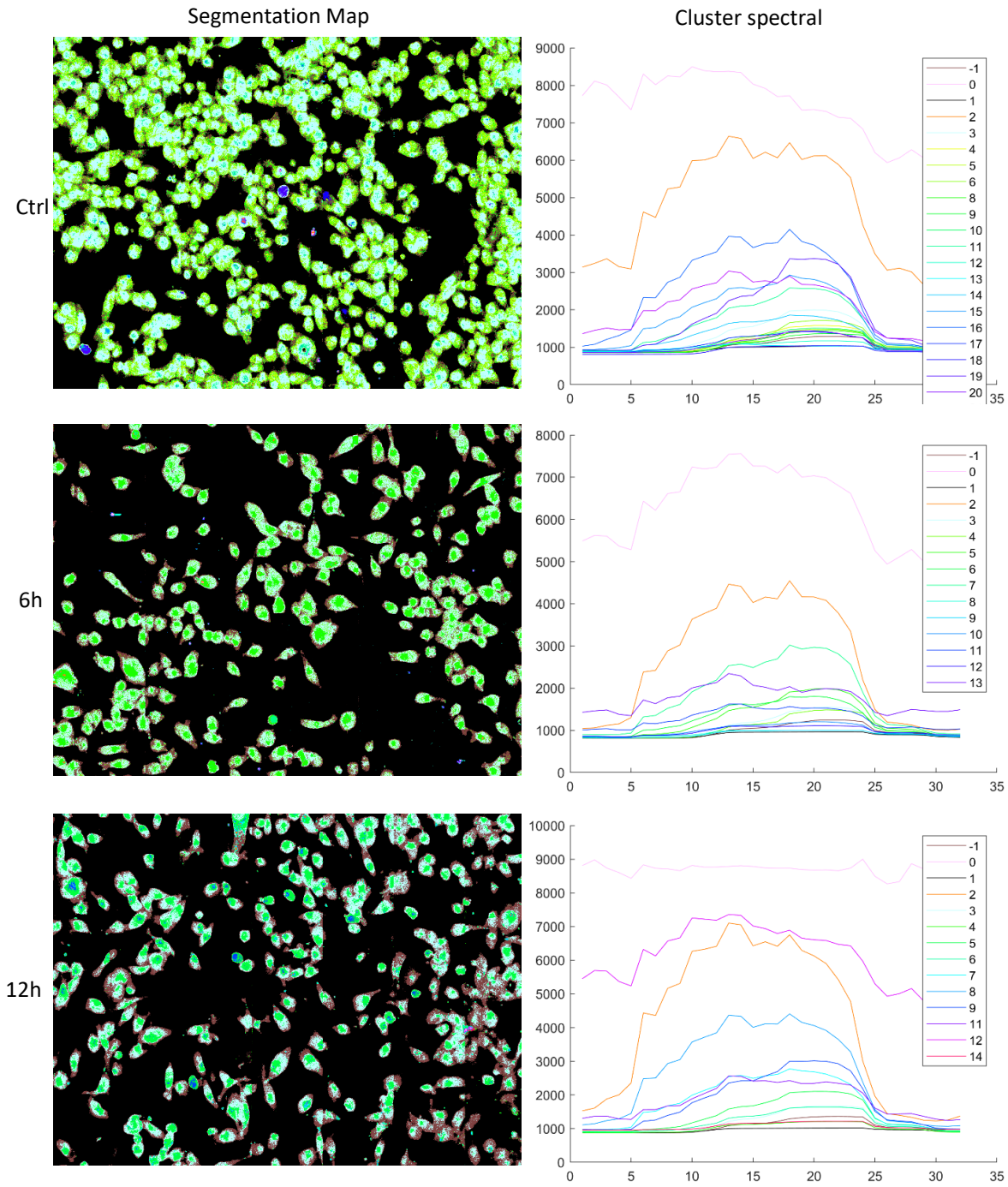


Figure 3.10. Segmentation map and corresponding spectral for proposed method.

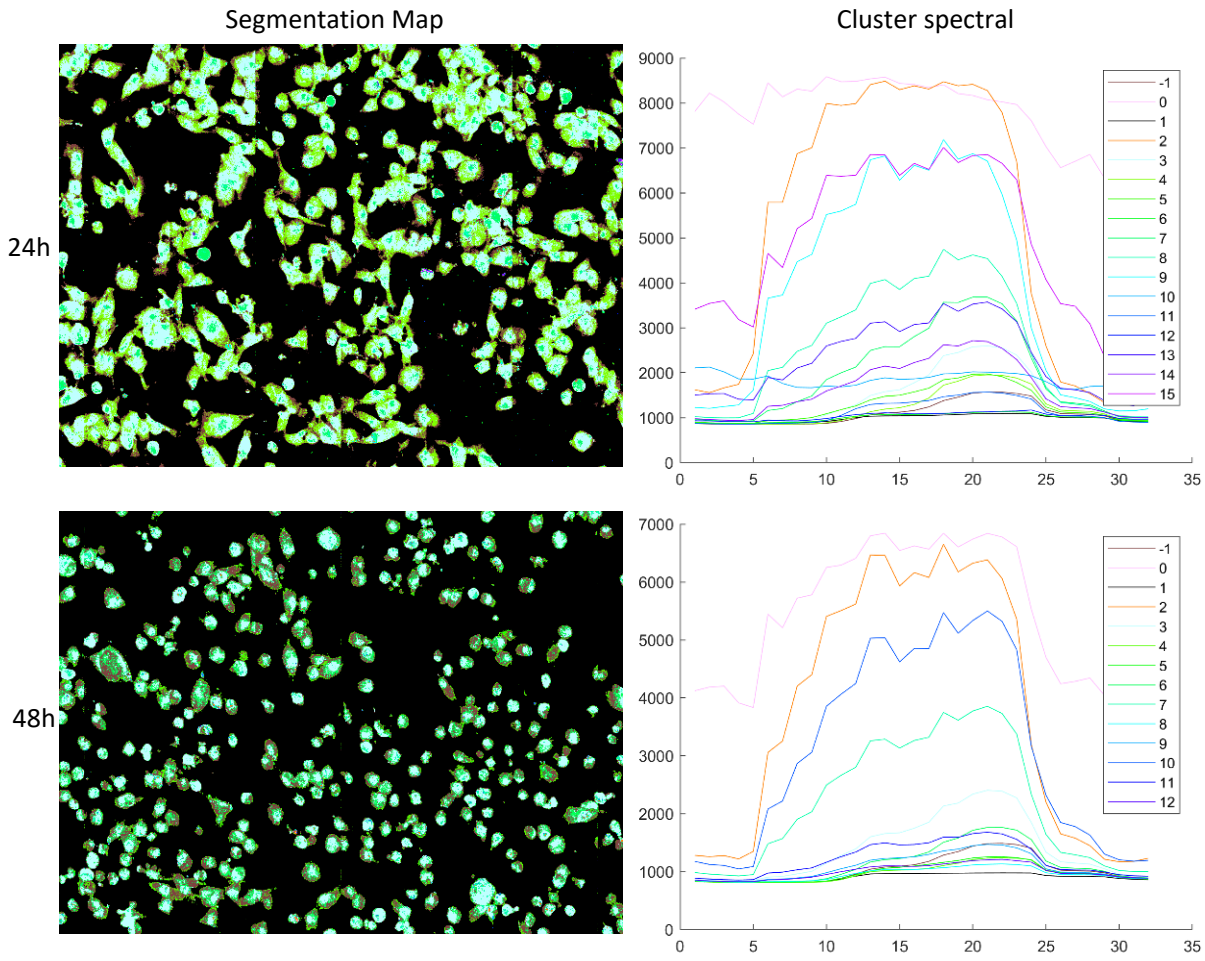


Figure 3.11. Segmentation map and corresponding spectral for proposed method continued.

3.3.5 Conclusions

In this work we propose a weakly-supervised deep non-exhaustive segmentation technique for segmentation of sub-cell structures in hyperspectral SRS images. Our approach starts the training process with coarse-grained labels obtained by simple PCA-based processing. As these coarse labels only provides "weak" supervision and identify obvious spectral patterns, the learning has to be performed so that class separability is preserved not only between these coarsely labeled classes but also between potential subclasses of these classes as well. To achieve this goal we train the network using the sum of reconstruction and discrimination losses. We replace the standard cross-entropy loss by a newly proposed contrastive loss function that is less sensitive to class labels than the cross-entropy loss so that in the learned feature space class separability information among unknown classes can be relatively well-preserved. Our contrastive loss function is based on the idea of maximizing the number of samples that falls on the right side of margin. Once the feature space is learned hyperspectral pixel data are projected into this space and clustered by a doubly non-parametric Bayesian clustering technique. This clustering technique can accommodate clusters with arbitrary shapes and automatically infers the number of classes from the data. Segmentation maps obtained using cluster labels for each pixel along with average spectra for each cluster suggest that our approach can discover a diverse set of biologically spectral patterns in SRS images with little supervision automaticaly obtained by PCA processing.

3.4 Applications to Non-Exhaustive Cell Segmentation with Electron Microscopy Images

In additional to the application of the hyper-spectral SRS image segmentation where the spatial correlation is merely auxiliary, we made an attempt on non-exhaustive semantic segmentation with Transmission Electron Microscopy (TEM) images. TEM images are high resolution grayscale images, the photo taken by TEM can provide details on structure of tissues, cells, organelles, and macromolecular complexes. In our application, we are trying to segment subpopulations of unmyelinated and myelinated fibers of neural pathways. See figure 3.12 for an example.

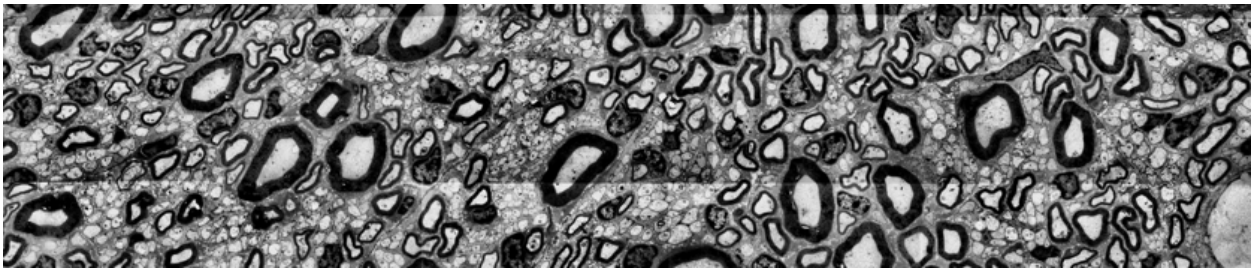


Figure 3.12. A crop of a TEM image after noise removal and histogram equalization.

It might be too early to talk about *non-exhaustive* semantic segmentation in general real-world scenarios, since scaling, obscuring, reflections, environmental effects etc. are still big challenges for semantic segmentation even when all class labels are existing. And these are challenges not only for semantic segmentation but also for the whole computer vision society. Without solving these problems, it is not possible for non-exhaustive semantic segmentation. But we still have hope on non-exhaustive semantic segmentation for the TEM images since it is much simpler. First, the cells are living in a 2D space, and there's not much environmental effects such as reflections, shadows etc. Second, the scaling problem is existed but much lighter. We can handle this through a multi-scale learning.

We explored semantic segmentation techniques that with success in semantic segmentation for biomedical images, and decided to develop our model based on Unet [66]. As a family of Fully Convolutional Network (FCN) [77] based architectures, Unet learns the segmentation mask in an end to end manner. Given an image patch with $W \times H$ pixels, it first did some down-sampling operations to learn a low resolution features which contains more global context information, and then perform the up-sampling operations taken into both the lower resolution features and the features from the down-sampling step with same resolution. Finally, a per-pixel label map of the same resolution of the input is learned by optimize a cross-entropy loss with regarding to the ground-truth label map.

The end to end training gives us convenience for non-exhaustive learning. Rather than training the network to generate a label map, we can train the network to generate a feature map, and then perform clustering on the generated feature map to infer the class label. A primitive approach to do this is to append a linear head before cross-entropy loss, and use the

output from the penultimate layer as the feature. Another alternative is to replace the cross-entropy loss with centroid margin loss proposed in 3.2. Though doing this gives us better performance than the primitive approach, we are still not satisfied by the results. During the experiments, we observed multi-modality in the background which made it the toughest class that contribute the most in the training. Therefore we proposed the Asymmetric Centroid Margin Loss in 3.4.1.

3.4.1 Asymmetric Centroid Margin Loss

It may not be a good idea to treat all class the same when there exists a class that represent something like irrelevant samples, artifacts, or background, which we called asymmetric class. These classes are usually multi-model since they are usually composed by sub-classes with large inter-class variation. For example background of the TEM images are usually multi-model with the existence of black or white crevice, some random texture changing etc. Therefore we proposed the Asymmetric Centroid Margin Loss (ACML) to handle these situations.

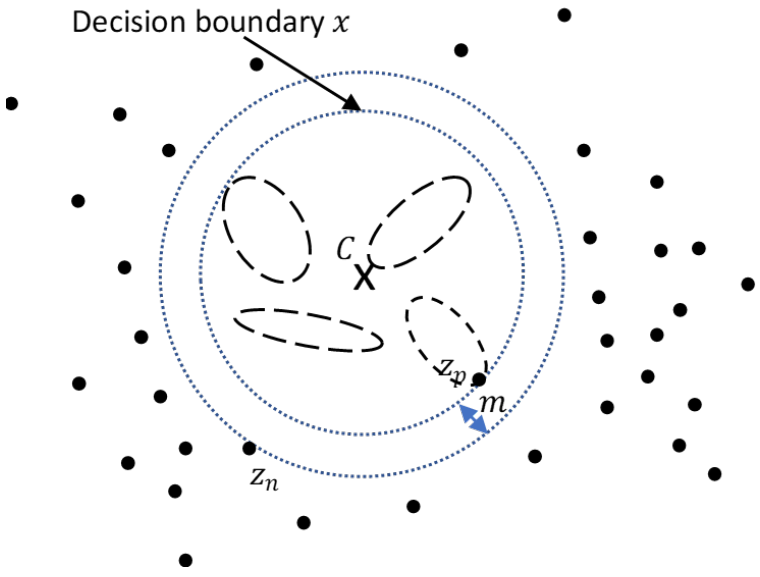


Figure 3.13. The illustration of asymmetric centroid margin loss.

The only different between CML and ACML is about how to treat with the asymmetric class. We first categorize samples in asymmetric class as negative samples, and samples

in target classes as positive samples. For positive samples we calculate normal CML as in equation 3.13. For negative samples, we calculate the loss based on the idea of pushing it to the peripheral of the overall positive samples distribution as illustrated in figure 3.13. In order to do this, we need found a decision boundary x which could best separate positive and negative samples. We did this by minimizing the number of miss-classified samples when predict samples with distance to the center of all positive samples C larger than x as negative and the samples with distance to C less than x as positive:

$$\max_{x} \sum_{p}^{N_p} \sigma\left(D(z_p, C) \leq x\right) + \sum_{n}^{N_n} \sigma\left(D(z_n, C) \geq x + M\right) \tag{3.17}$$

Where σ is the bool function equals 1 or 0 depending on whether the condition in the parenthesis is satisfied. After finding the decision boundary x through a dynamic programming algorithm based on sorted distance with computation complexity $O(n \log n)$, we calculate the loss as:

$$L = \sum_{p}^{N_p} \max\left(D(z_p, C) - x, 0\right) + \sum_{n}^{N_n} \max\left(x + M - D(z_n, C), 0\right) \tag{3.18}$$

where M is the margin.

3.4.2 Experiments

We performed experiments on a TEM image data set with 6 classes: background, unmyelinated fibers, outer myelinated fibers, inner myelinated fibers, blood vessels, and schwann cells. We select background, unmylinated fibers, outer myelinated fibers, and inner myelinated fibers as training classes, and only evaluate losses on pixels with training labels. We split the upper 50% part of the image as training, middle 10% as validation, and the rest bottom part as testing. An example of the original image (the validation part) is showed in figure 3.12.

We designed a data augmentation pipeline by randomly distorting the contrast, brightness, color inverse, and added some Gaussian random noises for a image patch. Each time we sampled a patch, it will have 50% possibility to be transformed. Note that the labels

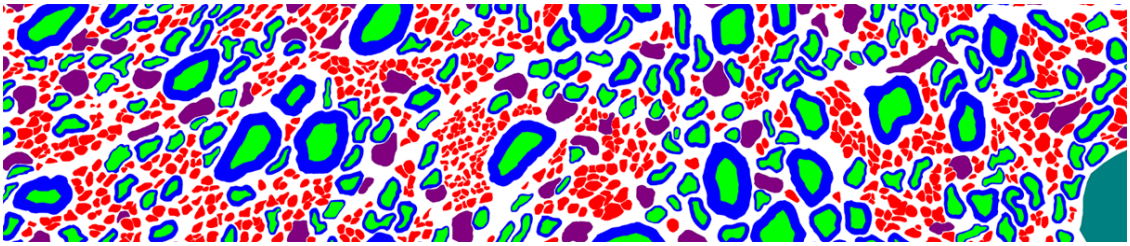
are the same for the transformed data, but the data transformed is very different from its original class, and we could not say it still belongs to its original class. This could cause trouble for cross-entropy loss but is not affect CML and its variants, because all the centers and label assignment are "local" within each training batch. Even if we change the labeling, as long as the modifications are consistent for all image patches within a training batch, it is not a problem for CML. We use the same U-Net backbone model and compared the results of learning 10 dimensional features through a cross-entropy loss, original CML, and ACML loss. For cross-entropy loss, we add a fully connected head, and obtained features from the output of the penultimate layer. In this way we can define the number of dimensions of features we want. Finally, we generate the label assignments by performing KMeans cluster on the learned feature, and then reshaped it back to compose a segmentation map.

Table 3.4. Comparison for TEM cell segmentation.

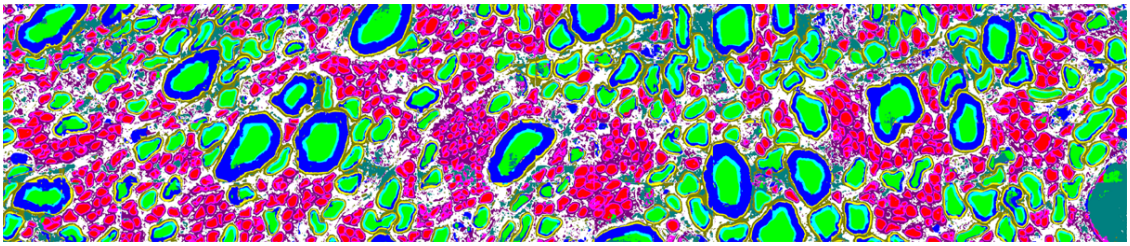
	cross-entropy loss	CML	ACML
macro F1	0.38	0.51	**0.64**
micro F1	0.40	0.60	**0.70**

The experiments results for testing data are showed in table 3.4. Where we evaluate the performance by micro and macro F1 scores defined in equation 3.16 and 3.15. We also plotted the segmentation map for the validation section of the image for ground-truth, CML result, and ACML result in figure 3.14.

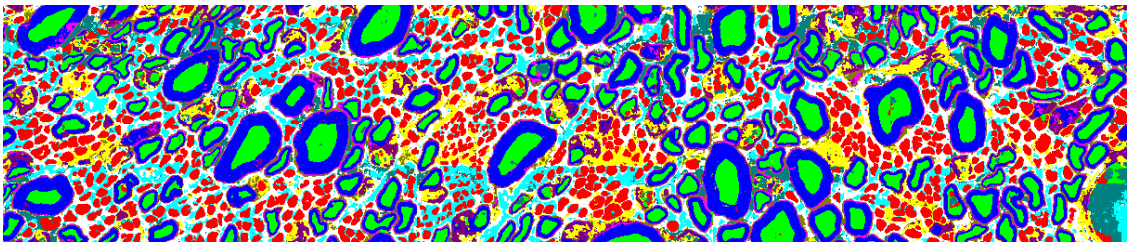
We can see from the results that ACML achieved the best F1 scores among the compared methods. And from the segmentation map we can see that it discovered some of the schwann cells (colored in purple). Though the discovery of schwann cell is not perfect as the model tend to split the border and the inner-side of the cell into two classes, the results is still much better than CML which classified most of the schwann cells as outer mylinated fibers. Moreover, we can see the effect of ACML that the background has been split into several subclasses, which could corresponding to different patterns of the background.



(a) The segmentation map for the ground truth.



(b) The segmentation map for CML results.



(c) The segmentation map for ACML results.

Figure 3.14. Comparison of the segmentation map. Where we use different colors for different classes.

4. OUT OF DISTRIBUTION DETECTION

To complete the story of learning in the open world, we studied the out-of-distribution detection (OOD) problem and considered ways to fuse it with non-exhaustive learning. OOD detection is proposed to enable the neural network classifier's deployment in real-world systems such as self-driving, bacteria detection etc, where unknown scenario can be emerging anytime. The need of OOD detection for our learning in the open world context is different from the need as traditional classifiers. Under the learning in the open world settings, the newly emerging classes is what we want to discover, but will be classified as OOD in the traditional classification problem. What we expect for OOD detection is to identify noises, anomalies, and outliers that hard to remove by reprocessing such as the CRISM case mentioned in section 1.1.

Recent works on OOD detection including classifier based methods which detect the OOD samples based on their classification likelihoods/scores [18], [78]; score based methods which design an objective as an indicator of how likely a sample is an OOD sample and later trained using neural networks [6], [79]; and distance based methods which determine whether a sample is an outlier based on some distance metrics between it and some anchors [17], [19]. We choose to conduct our research based on distance based methods since it is easy to integrate with our non-exhaustive learning models. Specifically, we explored some ideas based on the Mahalanobis OOD detector [19]. We described the details in 4.1 and illustrated with experiments on application to hyper-spectral mineral classification.

4.1 Mahalanobis Distance Based Score

We start by giving a review of the Mahalanobis OOD detector proposed by [19] below. In [19] the authors first train a deep neural network (DNN) with a softmax classifier to classifier the target dataset, then take the output from each of the intermediate layers until the penultimate layer of the trained DNN as the feature for OOD detection. Let denote the feature for layer l of data x to $f_l(x)$. For features from each layer, the sample mean for each class $\hat{\mu}_{l,c} = \frac{1}{N_c} \sum_{\mathrm{i}:y_\mathrm{i}=c} f_l(x_\mathrm{i})$ where N_c is the number of training samples with label c, and the tied sample covariance for the whole dataset $\hat{\Sigma}_l = \frac{1}{N} \sum_c \sum_{\mathrm{i}:y_\mathrm{i}=c} (f_l(x_\mathrm{i}) - \hat{\mu}_{l,c})(f_l(x_\mathrm{i}) - \hat{\mu}_{l,c})^T$ is

estimated and are used for calculating the Mahalanobis distance-based confidence score for layer l below:

$$M_l(x) = \max_c -(f_l(x) - \hat{\mu}_{l,c})^T \hat{\Sigma}_l^{-1} (f_l(x) - \hat{\mu}_{l,c}) \tag{4.1}$$

Which is the negation of the Mahalanobis distance without square root between the testing sample and its closest class in layer l. Then the final confidence score is calculated as a weighted average of the confidence score over all layers:

$$M(x) = \sum_l \alpha_l M_l(x) \tag{4.2}$$

Where the weights α_l are tuned by a logistic regression on a binary classification training dataset composed of in-distribution samples and OOD samples. Moreover, the authors also adopted a calibration technique same as in [18]. Rather than calculate the confidence score on the raw sample, the sample x are preprocessed by adding a small perturbation as below:

$$\hat{x}_l = x + \epsilon \operatorname{sign}(\nabla_x - M_l(x)) \tag{4.3}$$

where ϵ is a magnitude of noise also tuned on the binary classification training dataset with in-distribution samples and OOD samples. And the final $M_l(x)$ for each layer used to calculate the confidence score in (4.2) is actually calibrated as $M_l(\hat{x}_l)$. By adding a small perturbation of the gradient of the confidence score will make the in- and out-of-distribution samples more separable especially when their scores are close [18].

4.2 Applications to Hyper-spectral Mineral Classification

From the above description we can see that there's two place we can make improvement in the Mahalanobis OOD detector. First, the Mahalanobis OOD detector requires training with part of the ground-truth OOD samples, which is not practical in most of the real world applications. To deal with this, we generate simulated OOD samples and use the generated OOD samples to tune the layer ensemble weight α_l and the magnitude ϵ. We tested different sample generation methods and select the best performed one based on a

synthetic dataset. Then the same sample generation technique is used in the application. Second, tied covariance is used in the Mahalanobis OOD detector, whereas we compared the performance of both tied and un-tied covariance in our application. The un-tied covariance is calculated for each class c and per each layer l as $\hat{\Sigma}_{l,c} = \frac{1}{N_c} \sum_{\mathrm{i}:y_\mathrm{i}=c} (f_l(x_\mathrm{i}) - \hat{\mu}_{l,c})(f_l(x_\mathrm{i}) - \hat{\mu}_{l,c})^T$ where $\hat{\mu}_{l,c}$ is estimated same as above.

Before applying the algorithm to the real world hyper-spectral data, we first generated a synthetic data in the same domain to find a way that generate good simulated OOD samples. The synthetic data is generated by mixing library spectral [80] with noises added. The mixing weight for each data is sampled from a dirichlet prior. We categorize data with dominant mixing weight less than 0.5 as the OOD samples. We tried to generate simulated OOD samples by using random simulated signals; random uniformed noises in the range of the dataset; and background noises by averaging all training data together, and adding a Gaussian noises on the mixing. It turned out turning the model with both of the random simulated signals and the background noise gives the best performance.

We then compared the performance of Mahalanobis OOD detector with tied covariance and un-tied covariance with a CRISM dataset contains noises and outliers in Table 4.1. We first trained a classifier on the CRISM dataset using a modification of ResNet34 by replacing the 2D filter for images to 1D filters which convolution on spectral. Then fix the classifier and train the Mahalanobis OOD detector on the training data (in-distribution samples) and simulated OOD samples. Finally we generate the confident scores for each of the testing samples, and evaluate the performance by the area under curve (AUC) score calculated by varying the threshold of the confident score that separate OOD samples and in-distribution samples. We also compared the mean F1 score without OOD detection and with OOD detection. The mean F1 score is calculated as the average of F1 scores of in-distribution classes by treating OOD samples as a background class. If an OOD sample is assigned to a in-distribution class the FP of that class will be increase, hence reduce the F1 score. To calculate the mean F1 score with OOD detection, we chose the threshold based on the AUC curve that with max gain, and labeled the samples with confident score high than the threshold as OOD samples.

Table 4.1. Comparison of tied and un-tied covariance on CRISM

	mean F1 before/after noise detection	auc test
Mahalanobis OOD with tied covariance	0.797/0.801	0.822
Mahalanobis OOD with un-tied covariance	0.797/0.810	0.837

We can see from the results in Table 4.1 that the mean F1 score for both of the methods improved after OOD detection, and the improvement for un-tied covariance is larger than tied covariance, which indicated that using tied covariance is helpful in the hyper-spectral mineral classification application. To combine the Mahalanobis OOD detector with NEL algorithms is straight forward. Simply replace the classifier with the NEL model, and calculate the confidence score with regard to the clusters found by the NEL model. This is also a future work of our research.

5. CONCLUSION

In this research, we explored the problem of performing machine learning in an open world settings, where there could be things "unknown" happening. We first investigated ways to facilitate class discovery by learning from known classes. This problem,which we called Non-Exhaustive Learning (NEL), was the key component of our research. After initially approaching the problem from a generative model aspect, we concluded that a new feature space in which both known and unknown classes are well separated will be needed for complex structural real world datasets. We extended NEL work by developing Non-Exhaustive Feature Learning (NEFL) techniques that focused on feature learning . In addition to new class discovery we also explored out-of-distribution sample detection to complement our efforts in open world machine learning.

For NEL, we proposed the AI2GMM model 2.1 that simultaneously performs classification and new class discovery by inferring from sample statistics. AI2GMM is build upon a statistical model which is flexible to sample distribution, and is designed to adapt its hyper-parameters on both known and generated classes. Since hyper-parameters are used to guide the generation of new classes, AI2GMM showed its advantage on both synthetic and benchmark datasets compared to models with fixed hyper-parameters and models that assume Gaussian class distribution. But since label inference for AI2GMM relies on Markov Chain Monte Carlo (MCMC) sampling, it has high computational cost. There is a line of work that explores collapsed Gibbs sampling in order to achieve speed up by parallelization without losing much on performance [81]. Improving on this line of work could be one of future research directions.

As for the NEFL, we illustrated the utility of Centroid Margin Loss (CML) in terms of both the goodness of discrimination and generalization on unknown classes 3.2. Moreover, the Asymmetric Centroid Margin Loss (ACML) 3.4.1 could be naturally adopted for Out Of Distribution detection (OOD). In ACML we define samples that distribute outside of a hyper-spherical decision surface of interested classes and are multi-mode as the asymmetric class, which is actually the same as OOD samples. Instead of performing OOD by post-processing or pre-processing that are independent of the recognition, with ACML we can

perform OOD and NEL in a unified framework. But since ACML is a metric learning method which learns a feature space that does not generate labels by itself, in order to do this we need to develop an asymmetric label sampling model that could jointly model interesting distributions and aware of the peripheral samples. One naive idea to do this is to learn a threshold for AI2GMM while assigning a sample as a new cluster. If the posterior predictive distribution for the sample belongs to empty cluster higher than the threshold, then we make this assignment, otherwise we classify it as an OOD sample. This future extension will help us build a unified framework for open world machine learning.

REFERENCES

[1] B. Lake, R. Salakhutdinov, J. Gross, and J. Tenenbaum, "One shot learning of simple visual concepts," in *Proceedings of the annual meeting of the cognitive science society*, vol. 33, 2011.

[2] B. M. Lake, T. D. Ullman, J. B. Tenenbaum, and S. J. Gershman, "Building machines that learn and think like people," *Behavioral and brain sciences*, vol. 40, 2017.

[3] B. Romera-Paredes and P. Torr, "An embarrassingly simple approach to zero-shot learning," in *International Conference on Machine Learning*, 2015, pp. 2152–2161.

[4] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," in *Advances in neural information processing systems*, 2017, pp. 4077–4087.

[5] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, "A survey on deep transfer learning," in *International conference on artificial neural networks*, Springer, 2018, pp. 270–279.

[6] T. DeVries and G. W. Taylor, "Learning confidence for out-of-distribution detection in neural networks," *arXiv preprint arXiv:1802.04865*, 2018.

[7] W. J. Scheirer, A. de Rezende Rocha, A. Sapkota, and T. E. Boult, "Toward open set recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 7, pp. 1757–1772, 2012.

[8] Y.-C. Hsu, Z. Lv, and Z. Kira, "Learning to cluster in order to transfer across domains and tasks," *arXiv preprint arXiv:1711.10125*, 2017.

[9] K. Han, A. Vedaldi, and A. Zisserman, "Learning to discover novel visual categories via deep transfer clustering," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 8401–8409.

[10] F. Akova, M. Dundar, V. J. Davisson, E. D. Hirleman, A. K. Bhunia, J. P. Robinson, and B. Rajwa, "A machine-learning approach to detecting unknown bacterial serovars," *Statistical Analysis and Data Mining: The ASA Data Science Journal*, vol. 3, no. 5, pp. 289–301, 2010.

[11] M. Dundar, F. Akova, A. Qi, and B. Rajwa, "Bayesian nonexhaustive learning for online discovery and modeling of emerging classes," in *Proceedings of 29th International Conference on Machine Learning (ICML'12)*, Omnipress, 2012, pp. 113–120.

[12] Y. Cheng, B. Rajwa, and M. Dundar, "Bayesian nonparametrics for non-exhaustive learning," *arXiv preprint arXiv:1908.09736*, 2019.

[13] H. Z. Yerebakan, B. Rajwa, and M. Dundar, "The infinite mixture of infinite gaussian mixtures," in *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, 2014, pp. 28–36.

[14] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," in *Advances in neural information processing systems*, 2007, pp. 153–160.

[15] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[16] X. Ji, J. F. Henriques, and A. Vedaldi, "Invariant information clustering for unsupervised image classification and segmentation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 9865–9874.

[17] M. Radovanović, A. Nanopoulos, and M. Ivanović, "Reverse nearest neighbors in unsupervised distance-based outlier detection," *IEEE transactions on knowledge and data engineering*, vol. 27, no. 5, pp. 1369–1382, 2014.

[18] S. Liang, Y. Li, and R. Srikant, "Enhancing the reliability of out-of-distribution image detection in neural networks," *arXiv preprint arXiv:1706.02690*, 2017.

[19] K. Lee, K. Lee, H. Lee, and J. Shin, "A simple unified framework for detecting out-of-distribution samples and adversarial attacks," in *Advances in Neural Information Processing Systems*, 2018, pp. 7167–7177.

[20] F. Akova, D. Hirleman, A. K. Bhunia, B. Rajwa, and M. M. Dundar, "Non-exhaustive learning for bacteria detection," in *2009 International Conference on Network-Based Information Systems*, IEEE, 2009, pp. 206–211.

[21] E. Leask, B. Ehlmann, S. Murchie, and F. Seelos, "New possible crism artifact at 2.1 micrometers and implications for orbital mineral detections," *LPI*, p. 2840, 2018.

[22] F. Akova, M. Dundar, Y. Qi, and B. Rajwa, "Self-adjusting models for semi-supervised learning in partially observed settings," in *Data Mining (ICDM), 2012 IEEE 12th International Conference on*, IEEE, 2012, pp. 21–30.

[23] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei, "Sharing clusters among related groups: Hierarchical dirichlet processes," in *Proceedings of Advances in Neural Information Processing Systems*, 2005, pp. 1385–1392.

[24] N. Aghaeepour, G. Finak, H. Hoos, T. R. Mosmann, R. Brinkman, R. Gottardo, R. H. Scheuermann, F. Consortium, D. Consortium, *et al.*, "Critical assessment of automated flow cytometry data analysis techniques," *Nature methods*, vol. 10, no. 3, pp. 228–238, 2013.

[25] Y. Cheng, M. Dundar, G. Mohler, *et al.*, "A coupled etas-i2gmm point process with applications to seismic fault detection," *The Annals of Applied Statistics*, vol. 12, no. 3, pp. 1853–1870, 2018.

[26] Y. Ogata, "Statistical models for earthquake occurrences and residual analysis for point processes," *Journal of the American Statistical Association*, vol. 83, no. 401, pp. 9–27, 1988.

[27] Y. Ogata, "Space-time point-process models for earthquake occurrences," *Annals of the Institute of Statistical Mathematics*, vol. 50, no. 2, pp. 379–402, 1998.

[28] W. Spence, S. A. Sipkin, and G. L. Choy, "Measuring the size of an earthquake," *Earthquake Information Bulletin (USGS)*, vol. 21, no. 1, pp. 58–63, 1989.

[29] J. Gardner and L. Knopoff, "Is the sequence of earthquakes in southern california, with aftershocks removed, poissonian?" *Bulletin of the Seismological Society of America*, vol. 64, no. 5, pp. 1363–1367, 1974.

[30] T. Utsu, "A statistical study on the occurrence of aftershocks.," *Geophysical Magazine*, vol. 30, no. 4, pp. 521–605, 1961.

[31] A. Veen and F. P. Schoenberg, "Estimation of space–time branching process models in seismology using an em–type algorithm," *Journal of the American Statistical Association*, vol. 103, no. 482, pp. 614–624, 2008.

[32] D. Marsan and O. Lengline, "Extending earthquakes' reach through cascading," *Science*, vol. 319, no. 5866, pp. 1076–1079, 2008.

[33] J. Zhuang, Y. Ogata, and D. Vere-Jones, "Stochastic declustering of space-time earthquake occurrences," *Journal of the American Statistical Association*, vol. 97, no. 458, pp. 369–380, 2002.

[34] G. Adelfio and M. Chiodi, "Alternated estimation in semi-parametric space-time branching-type point processes with application to seismic catalogs," *Stochastic Environmental Research and Risk Assessment*, vol. 29, no. 2, pp. 443–450, 2015.

[35] I. Zaliapin, A. Gabrielov, V. Keilis-Borok, and H. Wong, "Clustering analysis of seismicity and aftershock identification," *Physical Review Letters*, vol. 101, no. 1, p. 018 501, 2008.

[36] A. Plesch, J. H. Shaw, C. Benson, W. A. Bryant, S. Carena, M. Cooke, J. Dolan, G. Fuis, E. Gath, and L. Grant, "Community fault model (cfm) for southern california," *Bulletin of the Seismological Society of America*, vol. 97, no. 6, pp. 1793–1802, 2007.

[37] G. O. Mohler, M. B. Short, P. J. Brantingham, F. P. Schoenberg, and G. E. Tita, "Self-exciting point process modeling of crime," *Journal of the American Statistical Association*, vol. 106, no. 493, pp. 100–108, 2011.

[38] J. Zhuang, "Next-day earthquake forecasts for the japan region generated by the etas model," *Earth, planets and space*, vol. 63, no. 3, p. 5, 2011.

[39] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

[40] M. Stephens, "Dealing with label switching in mixture models," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 62, no. 4, pp. 795–809, 2000.

[41] G. Mohler, "Marked point process hotspot maps for homicide and gun crime prediction in chicago," *International Journal of Forecasting*, vol. 30, no. 3, pp. 491–497, 2014.

[42] G. O. Mohler, M. B. Short, S. Malinowski, M. Johnson, G. E. Tita, A. L. Bertozzi, and P. J. Brantingham, "Randomized controlled field trials of predictive policing," *Journal of the American Statistical Association*, vol. 110, no. 512, pp. 1399–1411, 2015.

[43] E. Lewis and G. Mohler, "A nonparametric em algorithm for multiscale hawkes processes," *Preprint available at http://paleo.sscnet.ucla.edu/Lewis-Molher-EM_Preprint.pdf*, pp. 1–16, 2011.

[44] M. D. Porter and G. White, "Self-exciting hurdle models for terrorist activity," *The Annals of Applied Statistics*, vol. 6, no. 1, pp. 106–124, 2012.

[45] G. Mohler, "Modeling and estimation of multi-source clustering in crime and security data," *The Annals of Applied Statistics*, vol. 7, no. 3, pp. 1525–1539, 2013.

[46] G. White, M. Porter, and L. Mazerolle, "Terrorism risk, resilience, and volatility: A comparison of terrorism in three Southeast Asian countries," *Journal of Quantitative Criminology*, vol. 29, no. 2, pp. 295–320, 2013.

[47] G. White and M. Porter, "GPU accelerated MCMC for modeling terrorist activity," *Computational Statistics & Data Analysis*, vol. 71, pp. 643–651, 2014.

[48] E. Lai, D. Moyer, B. Yuan, E. Fox, B. Hunter, A. L. Bertozzi, and J. Brantingham, "Topic time series analysis of microblogs," DTIC Document, Tech. Rep., 2014.

[49] A. Simma and M. I. Jordan, "Modeling events with cascades of poisson processes," *arXiv preprint arXiv:1203.3516*, 2012.

[50] Q. Zhao, M. A. Erdogdu, H. Y. He, A. Rajaraman, and J. Leskovec, "Seismic: A self-exciting point process model for predicting tweet popularity," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, 2015, pp. 1513–1522.

[51] A. Khachaturyan, S. Semenovsovskaya, and B. Vainshtein, "The thermodynamic approach to the structure analysis of crystals," *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography*, vol. 37, no. 5, pp. 742–754, 1981.

[52] D. Dua and C. Graff, *UCI machine learning repository*, 2017. [Online]. Available: http://archive.ics.uci.edu/ml.

[53] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of eugenics*, vol. 7, no. 2, pp. 179–188, 1936.

[54] K. Pearson, "Liii. on lines and planes of closest fit to systems of points in space," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 2, no. 11, pp. 559–572, 1901.

[55] A. Coates and A. Y. Ng, "Learning feature representations with k-means," in *Neural networks: Tricks of the trade*, Springer, 2012, pp. 561–580.

[56] H.-C. Shin, M. R. Orton, D. J. Collins, S. J. Doran, and M. O. Leach, "Stacked autoencoders for unsupervised feature learning and multiple organ detection in a pilot study using 4d patient data," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1930–1943, 2012.

[57] Y. Bengio, A. C. Courville, and P. Vincent, "Unsupervised feature learning and deep learning: A review and new perspectives," *CoRR, abs/1206.5538*, vol. 1, p. 2012, 2012.

[58] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[59] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[60] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey, "Adversarial autoencoders," *arXiv preprint arXiv:1511.05644*, 2015.

[61] Z. Jiang, Y. Zheng, H. Tan, B. Tang, and H. Zhou, "Variational deep embedding: An unsupervised and generative approach to clustering," *arXiv preprint arXiv:1611.05148*, 2016.

[62] G. Chechik, V. Sharma, U. Shalit, and S. Bengio, "Large scale online learning of image similarity through ranking.," *Journal of Machine Learning Research*, vol. 11, no. 3, 2010.

[63] A. Kolesnikov, X. Zhai, and L. Beyer, "Revisiting self-supervised visual representation learning," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2019. DOI: 10.1109/cvpr.2019.00202. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2019.00202.

[64] J. Xie, R. Girshick, and A. Farhadi, "Unsupervised deep embedding for clustering analysis," in *International conference on machine learning*, 2016, pp. 478–487.

[65] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3213–3223.

[66] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, Springer, 2015, pp. 234–241.

[67] K. Sun, B. Xiao, D. Liu, and J. Wang, "Deep high-resolution representation learning for human pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5693–5703.

[68] J.-X. Cheng and X. S. Xie, "Vibrational spectroscopic imaging of living systems: An emerging platform for biology and medicine," *Science*, vol. 350, no. 6264, 2015.

[69] A. B. Hamida, A. Benoit, P. Lambert, and C. B. Amar, "3-d deep learning approach for remote sensing image classification," *IEEE Transactions on geoscience and remote sensing*, vol. 56, no. 8, pp. 4420–4434, 2018.

[70] H. Z. Yerebakan, B. Rajwa, and M. Dundar, "The infinite mixture of infinite gaussian mixtures," 2015.

[71] K. Zuiderveld, "Contrast limited adaptive histogram equalization," *Graphics gems*, pp. 474–485, 1994.

[72] J. S. Lim, "Two-dimensional signal and image processing," *Englewood Cliffs*, 1990.

[73] N. Otsu, "A threshold selection method from gray level histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, pp. 62–66, 1979.

[74] N. Audebert, B. Le Saux, and S. Lefevre, "Deep learning for classification of hyperspectral data: A comparative review," *IEEE Geoscience and Remote Sensing Magazine*, vol. 7, no. 2, pp. 159–173, 2019. DOI: 10.1109/MGRS.2019.2912563.

[75] R. R. Nidamanuri and B. Zbell, "Normalized spectral similarity score (NS^3) as an efficient spectral library searching method for hyperspectral image classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 4, no. 1, pp. 226–240, 2010.

[76] J. M. Nascimento and J. M. Dias, "Vertex component analysis: A fast algorithm to unmix hyperspectral data," *IEEE transactions on Geoscience and Remote Sensing*, vol. 43, no. 4, pp. 898–910, 2005.

[77] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[78] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *Advances in neural information processing systems*, 2017, pp. 6402–6413.

[79] J. Ren, P. J. Liu, E. Fertig, J. Snoek, R. Poplin, M. Depristo, J. Dillon, and B. Lakshminarayanan, "Likelihood ratios for out-of-distribution detection," in *Advances in Neural Information Processing Systems*, 2019, pp. 14 707–14 718.

[80] R. F. Kokaly, R. N. Clark, G. A. Swayze, K. E. Livo, T. M. Hoefen, N. C. Pearson, R. A. Wise, W. M. Benzel, H. A. Lowers, R. L. Driscoll, and A. J. Klein, *Usgs spectral library version 7*, 2017.

[81] H. Z. Yerebakan and M. Dundar, "Partially collapsed parallel gibbs sampler for dirichlet process mixture models," *Pattern Recognition Letters*, vol. 90, pp. 22–27, 2017.

VITA

Education

PhD student in Computer Science at Purdue University - West Lafayette, 2015 - present. Thesis title: "Machine Learning in the open world".

Master of Science in Computer Science at State University of New York at Buffalo, 2013 - 2015.

Bachelor of Science in Computer Science, East China University of Science and Technology, Shanghai, China, 2009-2013.

Work Experiences

Software engineering internship at Google - Apps Search Quality & Intelligence Team, Sunnyvale, May 2019 - August 2019. Project description: Multi-task sequential model for Gmail user modeling.

Software engineering internship at Google - Display Ads Quality and Format Team, Mountain View, May 2018 - August 2018. Project description: Unsupervised foreground and background segmentation for products images of ads auto-composition system.

Co-founder and Tech Leader at Snowberg Inc., Buffalo, NY (startup) April 2015 - July 2015.

Internship at SARI, Chinese Academy of Sciences April 2013 - June 2013.

Publications

- Cheng, Y.*, Qin, Z.*, et al. Multitask Mixture of Sequential Experts for User Activity Streams. 26th ACM SIGKDD, 2020. (* equal contribution)
- Cheng, Y., Rajwa, B., Dundar, M. Bayesian Nonparametrics for Non-exhaustive Learning. NIPS, Bayesian Nonparametrics Workshop, 2018.
- Cheng, Y., Dundar, M., Mohler, G. A Coupled ETAS-I2GMM Point Process with Applications to Seismic Fault Detection. Annals of Applied Statistics. 2018.