BAYESIAN METHODS AND MACHINE LEARNING FOR PROCESSING TEXT AND IMAGE DATA

by

Yingying Gu

A Dissertation Submitted in

Partial Fulfillment for the

Requirements for the Degree of

Doctor of Philosophy

in Engineering

at

The University of Wisconsin-Milwaukee

August 2017

ABSTRACT

BAYESIAN METHODS AND MACHINE LEARNING FOR
PROCESSING TEXT AND IMAGE DATA

by

Yingying Gu

The University of Wisconsin-Milwaukee, 2017
Under the Supervision of Professor Jun Zhang

Classification/clustering is an important class of unstructured data processing problems. The classification (supervised, semi-supervised and unsupervised) aims to discover the clusters and group the similar data into categories for information organization and knowledge discovery. My work focuses on using the Bayesian methods and machine learning techniques to classify the free-text and image data, and address how to overcome the limitations of the traditional methods. The Bayesian approach provides a way to allow using more variations(numerical or categorical), and estimate the probabilities instead of explicit rules, which will benefit in the ambiguous cases. The MAP(maximum a posterior) estimation is used to deal with the local maximum problems which the ML(maximum likelihood) method gives inaccurate estimates. The EM(expectation-maximization) algorithm can be applied with MAP estimation for the incomplete/missing data problems. Our proposed framework can be used in both supervised and unsupervised classification. For natural language processing(NLP), we applied the machine learning techniques for sentence/text classification. For 3D CT image segmentation, MAP EM clustering approach is proposed to auto-detect the number of objects in the 3D CT luggage image, and the prior knowledge and constraints in MAP estimation are used to avoid/improve the local maximum problems. The algorithm can automatically determine the number of classes and find the optimal parameters for each class. As a result, it can automatically detect the number of objects

and produce better segmentation for each object in the image. For segmented object recognition, we applied machine learning techniques to classify each object into targets or non-targets. We have achieved the good results with 90% PD(probability of detection) and 6% PFA(probability of false alarm). For image restoration, in X-ray imaging, scatter can produce noise, artifacts, and decreased contrast. In practice, hardware such as anti-scatter grids are often used to reduce scatter. However, the remaining scatter can still be significant and additional software-based correction are desirable. Furthermore, good software solutions can potentially reduce the amount of needed anti-scatter hardware, thereby reducing cost. In this work, the scatter correction is formulated as a Bayesian MAP (maximum a posteriori) problem with a non-local prior, which leads to better textural detail preservation in scatter reduction. The efficacy of our algorithm is demonstrated through experimental and simulation results.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

First and foremost, I would like to express my sincere thanks to my PhD advisor, Dr. Jun Zhang for all the guidance, support and encouragement he has given me during my time at University of Wisconsin-Milwaukee. I have been extremely fortunate to work in his research group. I sincerely appreciate his patience, help and support in my academic career.

Beside my advisor, I would like to thank the rest my dissertation committees: Dr. Susan W McRoy, Dr. Tian Zhao, Dr. Yi Hu, Dr. Seyed H Hosseini, Dr. Daniel Gervini, and Dr. Yirong Wu. Thank you for their encouragement, help and insightful comments.

Furthermore, I would like to thank our past and present group members: Dr. Weisong Liu, Dr. Jun Wang, Hongquan Zuo, Christopher Kallas, Ivko Cvejic, Wen Hu, and Ying Zhu. Thank you for your friendships, good advice and collaborations.

Lastly, I would like to thank my parents for all their love and encouragement, and my husband Lin Liu who has been hugely supportive all these years. I am very grateful. Thank you.

Chapter 1

Introduction

We are in an era of "big data". According to International Data Corporation(IDC) analysis, digital data is growing dramatically from 4 zettabytes (ZB, 10^{21}bytes) in 2013 and will double in size every two years, up to 44 zettabytes by 2020. The "big data" refers to both the structured and unstructured data. The structured data is tagged and can be saved in the relational database. On the contrary, the unstructured data is referred to free-text (emails, blogs, websites, medical reports, etc.), images, audio and video, etc, and cannot fit in a traditional database. According to the 2013 IDC analysis, only 22% of the digital data was structured/tagged. This data could have been useful to analyze, but less than 5% was actually analyzed.

The "big data", especially for the high proportion (∼80%) of unstructured data, is underestimated because there is a lack of technology for processing unstructured data. First, the important information is hidden or not labeled in these unstructured data. Second, the unstructured data(free-text, image, and video), often has many variations(numerical or categorical), which makes the recognition very difficult. Third, the data is in a huge volume and growing extremely fast. There are many benefits if we can analyze and control the big data. By using the real-time data, a business can make decisions faster and increase productivity and profitability more than competitors. The

data from healthcare is also enormous and growing exponentially every year. Gaining control and insight of the high-value data could improve the qualities of patient care. Classification is an important class of unstructured data processing problems. The classification aims to discover the clusters and group similar data into categories for information organization and knowledge discovery. There are two main types of methods. One is the supervised classification approach. The goal of this approach is to predict/classify new data with the labeled training data available. The other is unsupervised clustering with unknown knowledge of the data. The objective is to discover the hidden structures of the observations and cluster the data into groups based on the similarities. The unsupervised clustering is more difficult, since there are unknown classes and no error calculations to use. On the contrary, the supervised classification could be compared with the observed/labled information.

There are many classification techniques, such as the commonly used centroid-based (K-means) clustering [1], connectivity-based (hierarchical) clustering [2, 3], distribution-based (such as multivariate normal distribution) clustering [4, 5], and density-based clustering [7, 8]. Although these classification techniques are widely used, they all have drawbacks. First, the problem with the conventional clustering method is that we need to know the proper number of clusters before the clustering algorithm, and there is no reliable way to know it in reality. Second, most of the optimization algorithms do not guarantee to find the global optimal solution. It only improves upon the initial set of parameters that are chosen randomly or from any guesses, and returns the sub-optimal solutions. This is known as local maxima problems. Third, due to the curse of high-dimensionality [9], most of the methods will fail in high-dimensional variables. My work focuses on using the Bayesian model-based techniques for classification and clustering on unstructured data (free-text and images), and addressing to improve the limitations of the traditional clustering methods.

The Bayesian model-based techniques have the following advantages: First, it allows to

use more variations, such as continuous(numerical) or discrete (binary or categorical) numbers, or intensity values of image, etc. We will use the well-known exponential-family distribution models [19]. For discrete data, which is often represented in text classification, the Bernoulli model and multinomial is used. The continuous data is usually observed in the signal and image processing applications, and the Gaussian distribution is popularly used. Second, in the ambiguous cases, it is better to use the probability estimations instead of explicit decision rules for the predictions. The model's decisions are based on a specific likelihood threshold, which can change/optimize the recall and precision. Third, it allows maximum use of prior knowledge and constraints in the probability estimation, which is known as MAP(maximum a posteriori) estimate. The MAP estimation is used to deal with the local maximum problems which the ML(maximum likelihood) method gives inaccurate estimates. Fourth, The EM(expectation-maximization) algorithm[13] can be applied with MAP estimation for the incomplete/missing data problems. For general EM algorithm which is used with ML estimation for incomplete data, it suffers the local maximum problems, especially for the high dimension data. By using MAP, it could avoid/improve the local maximum problems by adding the prior knowledge/constraints. In this thesis, we will use the Bayesian classification techniques on two real-world applications.

First, supervised classification is widely used to solve many real-world problems when we have some training examples available. We used it to automatically identify qualified patients for breast cancer clinical trials from free-text medical reports. Currently, at most participating hospitals, qualified patients are identified manually by research nurse coordinators who review patient medical reports, which are usually free text unstructured documents, to identify the patients that meet the criteria of a particular trial. The process of identifying enough patients for a clinical trial is costly, inefficient, and labor-intensive. As a result, some trials may be delayed. The purpose of our work is to develop algorithms to automatically identify qualified patients for breast cancer

clinical trials from free-text medical reports. Our new approach uses a trained Bernoulli Model which is created from our training data (qualified or unqualified sentences) [41]. The trained Bernoulli model calculates the probability of a new word sequence (test sentence) to make a classification decision. Our method has several advantages. There is no required special pre-processing, such as syntactic parsing, POS tagging etc. This is beneficial for the free-form text medical report, which may contain grammatical mistakes during transcription. The algorithm is fast, scalable, and could be easily applied to any natural language. The model's decisions are based on a specific likelihood threshold, which can change/optimize the recall and precision.

Second, unsupervised clustering is used to separate the data into different groups based on the similarity. Unlike the supervised classification, the unsupervised Bayesian clustering is used to discover the clusters without any training information available. Today, CT security screening is highly used in airports or other security places to detect if there are any threats in the luggage. It is important for preventing the dangerous objects in transportation. However, it has very high false alarms rates because some non-threatening objects share similar characteristics with the actual threats. Also, unavoidable metal artifacts are often in CT luggage images. If any suspicious objects are detected, the luggage has to be manually verified. Because of the high false alarm rates, the processing of manual verification is inefficient and costly. As a result, it delays the passengers and transportation of the luggage. Our work is to develop algorithms which could improve the threats detection rate and also reduce the false alarm rates for CT luggage screening. We proposed the MAP EM clustering method[4, 5, 14, 15, 16] to auto-detect the number of objects in the 3D CT luggage image. Our methods have several advantages. The prior knowledge and constraints can be used to avoid/improve the local maximum problems. The algorithm can automatically determine the number of clusters and find the optimal parameters for each class. As a result, it can automatically detect the number of objects and produce better segmentation for each

object in the image. After image segmentation process, we applied the supervised machine learning techniques for object classification.

Third, Bayesian MAP method can be used for scatter correction in X-ray imaging. Specifically, scatter correction amounts to estimating the original image when it is corrupted by Poisson noise and is also possibly distorted by a linear or a non-linear transformation, while image restoration amounts to estimating the original image when it is corrupted by Gaussian noise and is also possibly distorted by (usually) a linear transformation. Indeed, the aforementioned Total Variation(TV) and Markov Random Field(MRF) based techniques were initially developed for image restoration. Recently, a number of non-local techniques such as [58, 60] and their extensions have significantly improved image restoration results. Specifically, they provide better results by preserving textural details better, which is difficult by traditional or even TV and MRF techniques which are local in nature. Furthermore, recent advances in blind deconvolution (de-blurring) e.g., [62, 63] can potentially be exploited to produce better scatter model estimation. In this work, we adapt some of the non-local techniques for image de-noising and image restoration to scatter reduction and demonstrate their efficacy. In addition to X-ray imaging, our techniques could also be extended relatively straightforwardly to scatter correction in CT (computer tomography).

The contributions of my work are:

- Applied the Bayesian approach and machine learning techniques to medical free-text reports:
  - Proposed the probability models for text mining
  - Applied machine learning techniques for automatic patient search
- Applied the Bayesian approach to 3D CT image segmentation and machine learning techniques to classification

  - Proposed the MAP EM GMM(Gaussian mixture models) for automatically detecting the objects in 3D CT image
  - Developed a pixel-based MRF(Markov Random Field) segmentation algorithm
  - Applied machine learning classification for object classification

- Applied the Bayesian approach to X-ray image restoration
  - Proposed the scatter models and Poisson image model
  - Developed Bayesian MAP with non-local means prior for scatter correction

This dissertation is organized as follows: Chapter 2 is a review of related work. Chapter 3 describes our supervised classification method on medical free-text reports. Chapter 4 shows our proposed approach of unsupervised clustering technique. Chapter 5 discusses objects detection and recognition in 3D CT luggage image. Chapter 6 shows our scatter correction method for X-ray image by using Non-Local techniques.

Chapter 2

Related Work

Many Bayesian classification methods have been proposed to solve problems in different areas. In this section, we will discuss some well-known Bayesian related methods.

2.1 Exponential-Family Distribution

In reality, our observation data is either categorical or numerical. Model selection is to select the right distribution model according to the observations/given data. In this thesis, we used the following probability distribution models, which are from the well-known exponential-family [19].

2.1.1 Multivariate Normal(Gaussian) Distribution

Gaussian distribution is the most popular density probability function to model the continuous or numerical data. The d dimensions of Gaussian density function can be defined as:

$$p(x|\mu,\Sigma) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left\{-\frac{1}{2}(x-\mu)^T(\Sigma)^{-1}(x-\mu)\right\} \tag{2.1}$$

, where $x \in \mathbb{R}^n$, $\mu \in \mathbb{R}^n$, $\Sigma \in \mathbb{R}^{n \times n}$. Suppose an i.i.d. data set $D = \{x_1, x_2, ..., x_n\}$ comes from a mixture of K-class Gaussian distribution $N(\mu_j, \Sigma_j)$ with mixture weight/proportion of π_j for each class. So the K-class mixture density can be written as:

$$p(x_i|\boldsymbol{\theta}) = \sum_{j=1}^{K} \pi_j p\left(x_i|\mu_j, \Sigma_j\right) \tag{2.2}$$

, where $\boldsymbol{\theta} = \{\mu_1, \mu_2, ..., \mu_K; \Sigma_1, \Sigma_2, ..., \Sigma_K; \pi_1, \pi_2, ..., \pi_K\}$ and $\sum \pi_j = 1$, $\pi_j > 0$.

2.1.2 Inverse-Wishart Distribution

The inverse-Wishart distribution is used as the conjugate prior for covariance matrix of a multivariate Gaussian distribution, denoted as $W^{-1}(\Sigma|S_0, \nu_0)$. The probability density function of the inverse-Wishart is defined as:

$$W^{-1}(\Sigma|S_0, \nu_0) = \frac{|S_0|^{\nu_0/2}}{2^{\frac{\nu_0 d}{2}} \Gamma_d(\frac{\nu_0}{2})} |\Sigma_j|^{-(\nu_0+d+1)/2} \exp\left\{-\frac{1}{2} Tr(S_0 {\Sigma_j}^{-1})\right\} \tag{2.3}$$

, where Σ and S_0 are positive definite matrices, v_0 is degrees of freedom, Γ_d is the multivariate Gamma function, and $Tr(\cdot)$ is the trace of the matrix.

2.1.3 Bernoulli Distribution

Suppose the observation data is a binary sequence $\boldsymbol{S} = \{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$, where the $x_i \in [0, 1]$ represents the y_i observation occurs or not. The probability density function of the multivariate Bernoulli can be written as:

$$p(\boldsymbol{S}) = \prod_{i=1}^{n} p(y_i)^{x_i} (1 - p(y_i))^{1-x_i} \tag{2.4}$$

, where the $p(y_i)$ represents the probability of observation y_i in the given data set.

2.1.4 Beta Distribution

The beta distribution is a continuous distribution and parameterized by two parameters, which can determine the shape of distribution, denoted as $B(\alpha, \beta)$. The beta distribution is the conjugate prior probability distribution of Bernoulli, binominal and geometric distributions. The probability density function for beta distribution is:

$$p(x; \alpha, \beta) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1} = \frac{1}{B(\alpha, \beta)} x^{\alpha-1}(1-x)^{\beta-1} \tag{2.5}$$

, where $\Gamma(\cdot)$ is the gamma function, $B(\alpha, \beta)$ is beta function, which is also the normalization constant.

2.1.5 Dirichlet Distribution

The Dirichlet distribution is a continuous probability distribution parameterized by a parameter α, denoted as $Dir(\alpha)$. The Dirichlet distribution is the conjugate prior of categorical and multinomial distribution. The probability density function for Dirichlet distribution is:

$$p(x_1, x_2, ..., x_{K-1}; \alpha_1, \alpha_2, ..., \alpha_K) = \frac{1}{B(\alpha)} \sum_{i=1}^{K} {x_i}^{\alpha_i - 1} \tag{2.6}$$

, where x_i is a random variables, $x_i > 0$ and $\sum x_i = 1$, $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, ..., \alpha_K)$ are K positive numbers, and beta function $B(\cdot)$ is the normalization constant which can written by using gamma function as:

$$B(\boldsymbol{\alpha}) = \frac{\prod_{i=1}^{K} \Gamma(\alpha_i)}{\Gamma(\sum_{i=1}^{K} \alpha_i)} \tag{2.7}$$

From this equation, we can see that the Dirichlet distribution is to use a distribution parameterized by $\boldsymbol{\alpha}$ on the random variables $\boldsymbol{x}$.

2.2 Bayesian Analysis

Bayesian method was first proposed by Bayes and Laplace in 18th century, and it is widely used until in 20th century when the powerful computers are available. The fundamental theory is Bayes' Theorem. It is defined as the following form:

$$p(\theta_k|x) = \frac{p(x|\theta_k)p(\theta_k)}{\int p(x|\theta)p(\theta)d\theta} \tag{2.8}$$

, where $p(\theta|x)$ is the posterior probability, $p(x|\theta)$ the likelihood probability and $p(\theta)$ the prior probability, and the denominator is the marginal likelihood. The denominator is known as evidence normalization constant which could be ignored. So the posterior is proportional to the likelihood times the prior.

2.3 Maximum Likelihood (ML) Estimation

Now, we describe the definition of the maximum likelihood estimation method [10, 11] which is used to estimate the values of parameters based on the observations. Suppose we have a data set of n independently and identically distributed (i.i.d.) observations $\boldsymbol{x} = \{x_1, x_2, ..., x_n\}$ from a distribution of $p(x_i|\theta)$. The θ is the parameter of probability model which can be scalar or vector. Since the sample data are independent to each other, then, we can have

$$p(\boldsymbol{x}|\boldsymbol{\theta}) = \prod_{i=1}^{n} p(x_i|\theta) \tag{2.9}$$

. The $p(\boldsymbol{x}|\boldsymbol{\theta})$ is called the likelihood function of $\boldsymbol{\theta}$ with input data $\boldsymbol{x} = \{x_1, x_2, ..., x_n\}$. Since the **log** is a strictly monotonically increasing, the maximum likelihood estimation of $\boldsymbol{\theta}$ is usually calculated by maximizing the **log-ML** estimation as the following

formula:

$$\hat{\boldsymbol{\theta}}_{\mathrm{ML}} = \arg\max_{\boldsymbol{\theta}} \left\{ \sum_i \log p(x_i|\boldsymbol{\theta}) \right\} \tag{2.10}$$

For some applications, maximum likelihood estimation can be easily computed if it has an explicit form. However, if there is no explicit form which involves the latent variables or missing values among the data, ML estimation has to be solved numerically using optimization method such as Expectation-Maximization algorithm or some other techniques [13, 14, 15].

2.4 Maximum A Posterior (MAP) Estimation

Maximum a Posteriori (MAP) [12] is an estimation method to calculate the posterior distribution. Now, if we have the prior distribution on $\boldsymbol{\theta} = \{\theta_1, \theta_2, ..., \theta_K\}$. Then, we applied the Bayes' theorem on it and get the posterior distribution of $\boldsymbol{\theta}$ as:

$$p(\theta_k|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\theta_k)p(\theta_k)}{\int p(\boldsymbol{x}|\theta_i)p(\theta_i)d\theta} \tag{2.11}$$

So the MAP estimation of $\boldsymbol{\theta}$ is to maximize the posterior distribution with respect to θ. Because the denominator do not depend on θ, it can be treated as constant.

$$\hat{\boldsymbol{\theta}}_{\mathrm{MAP}} = \arg\max_{\boldsymbol{\theta}} \left\{ \frac{p(\boldsymbol{x}|\theta_k)p(\theta_k)}{\int p(\boldsymbol{x}|\theta_i)p(\theta_i)d\theta} \right\} = \arg\max_{\boldsymbol{\theta}} \{p(\boldsymbol{x}|\theta_k)p(\theta_k)\} \tag{2.12}$$

The more convenient form is to take the logarithm of the function as the following:

$$\hat{\boldsymbol{\theta}}_{\mathrm{MAP}} = \arg\max_{\boldsymbol{\theta}} \left\{ \sum_i \log p(x_i|\boldsymbol{\theta}) + p(\boldsymbol{\theta}) \right\} \tag{2.13}$$

From the formula, we can see that if the prior distribution of $p(\boldsymbol{\theta})$ is a uniform distribution or a constant, the MAP estimation can be reduced to the ML estimation. So, the MAP estimation could have better performance than ML estimation, in the

sense that the MAP estimation includes the prior knowledge/constraints which can be provided from training data and influences the estimation of $\boldsymbol{\theta}$. So, the next problem is about how to choose a proper prior distribution.

Also, MAP estimation can be easily computed in an explicit form when conjugate priors [20] are used. In other applications, if priors' distribution does not have an analytic form, it still can be solved by numerical optimizations, expectation-maximization or Monte Carlo technique [13, 14, 15].

2.5 General Expectation-Maximization (EM)

In reality, when our observation data is incomplete or has missing latent class/variables, Expectation-Maximization (EM) algorithm [4, 5, 13, 14, 15, 16] is proposed to deal with this problem. EM algorithm is an iterative ML procedure for parameters estimation in incomplete data or hidden parameters of the data.

We assume that the x is the complete data, and $x = \{y, z\}$, where y is the observation data, and z is the indicator that which class the y belongs to. For the incomplete data, when the z is unknown, and only y is remained. We have defined the likelihood function on complete data as

$$\mathcal{L}(\boldsymbol{\theta}; x_1, x_2, ..., x_n) = p(\boldsymbol{x}|\boldsymbol{\theta}) \tag{2.14}$$

The General EM has two steps. The E-step is to compute the expectation of log-ML for incomplete data as

$$Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) = E[\log p(\boldsymbol{x}|\boldsymbol{\theta})|y, \boldsymbol{\theta}^{(t)}] \tag{2.15}$$

, where $E[\cdot]$ is the expectation operator, y is the observation data, $\boldsymbol{\theta}^{(t)}$ is the current parameters. The M-step is to re-estimate the $\boldsymbol{\theta}$ by maximizing

$$\boldsymbol{\theta}^{(t+1)} = \arg\max_{\boldsymbol{\theta}} \left\{ Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) \right\} \tag{2.16}$$

The EM algorithm starts with randomly chosen parameters, and then iteratively repeated the E-step and M-step. The EM iteration stops/converges when the difference of the likelihood probability between two successive iterations is smaller than a threshold.

The drawback of conventional EM algorithm is that it is the local optimization method and hard to reach the global maxima. There are some proposed methods to avoid the local maxima [21, 22, 23, 24]. For example, EM is run in many times with different initializations, and kept the highest maximization estimation results. But this method is computationally expensive in the case of the convergence of EM is also slow. Another proposed method is to use greedy EM algorithm [24, 25]. The algorithm starts with a single class, and a new class is iteratively added until reaching the convergence criteria.

2.6 Total Variation Regularization

Let u be the noise-free image, y be the observed image, n be Gaussian noise, and k be the filter that models blurry and noisy image, we have

$$y = k * u + n = Ku + n \tag{2.17}$$

The total variation regularization can be used for de-noising and de-convolution applications as a prior. For example, it can be used in a Maximum A Posterior (MAP) approach for denosing and deconvolution:

$$u^* \in \arg\min_u ||y - Ku||^2 + \lambda J(u) \tag{2.18}$$

$$E(u) = \lambda \int ||y - k * u||^2 + J(u) \tag{2.19}$$

where u^* is the denoised and deblurred image and $J(u)$ is the total variation with

$$J(u) = \sum ||\nabla u|| \tag{2.20}$$

or

$$J(u) = \sum \sqrt{||\nabla u||^2 + \epsilon} \tag{2.21}$$

2.7 Gradient Descent Method

Gradient Descent is an iterative optimization method to find the minimum of a function. We can apply it to solve the inverse problem of de-blurring and de-noising image with total variation regularization.

$$u^{(i+1)} = u^{(i)} - \tau \left(k * (k * u^{(i)} - y) + \lambda GradJ(u^{(i)}) \right) \tag{2.22}$$

Where,

- τ is a step size.
- The gradient of the TV term is:

$$GradJ(u) = -div \left(\frac{\nabla u}{\sqrt{||\nabla u||^2 + \epsilon}} \right) \tag{2.23}$$

  - The gradient of u can be computed as

$$\nabla u = [u_x, u_y] \tag{2.24}$$

$$u_x = u_{i+1,j} - u_{i,j}$$

$$u_y = u_{i,j+1} - u_{i,j}$$

– $||\cdot||$ is the Euclidean Norm:

$$||\nabla u|| = \sqrt{u_x^2 + u_y^2} \tag{2.25}$$

– *div* is divergence:

$$div(u) = \frac{\partial u_x}{\partial x} + \frac{\partial u_y}{\partial y} \tag{2.26}$$

$$(div(u))_{i,j} = \begin{cases} u_x(i,j) - u_x(i-1,j), & \text{if 1<i<n;} \\ u_x(i,j), & \text{if i=1;} \\ -u_x(i-1,j), & \text{if i=n.} \end{cases}$$

$$+ \begin{cases} u_y(i,j) - u_y(i,j-1), & \text{if 1<j<n;} \\ u_y(i,j), & \text{if j=1;} \\ -u_y(i,j-1), & \text{if j=n.} \end{cases}$$

Chapter 3

Supervised Classification on Free Text

Hospitals kept lots of valuable electronic medical records from the treatment of the patients every day. Gaining the control and insights of the high-value medical data can improve the patient care qualities. However, most of the medical reports are in unstructured form which is hardly to be understood and analyzed except the doctors.

3.1 Motivation

For example, clinical trials are important to breast cancer research and treatment. For the results to be scientifically valid and clinically useful, a trial has to enroll a sufficient number of qualified patients. However, most the clinical information are stored in the free text form: surgeon visit notes, lab reports, and pathology reports. Currently, at most participation hospitals, qualified patients are identified manually by research nurse coordinators who review patient medical reports, which are usually free text unstructured documents, to identify the patients that meet the criteria of a particular trial. Since the number of reports associated with even a single patient can be quite large, including 5 or more surgeon visit notes, 5 or more lab reports, and several pathology reports, the process of identifying enough patients for a clinical trial is costly,

inefficient, and labor-intensive. As a result, some trials maybe delayed. Furthermore, not being identified, many qualified patients (as many as 60% according to some reports [27]) miss trials that can potentially benefit them.

The purpose of our work is to develop algorithms to automatically identify qualified patients for breast cancer clinical trials from free-text medical reports. The resulting software could be used in two ways. First, it could be used in an "interactive" mode by a research nurse coordinator to search for qualified patients (she can make the final decision based on the search results). Second, it could be used in a "background mode" to periodically scan patient reports and alert physicians when their patients qualify for various clinical trials.

3.2 Related Work

Despite its importance, it seems that the problem of automatically identifying qualified patients has not received sufficient attention. Indeed, a literature search in this area usually produces only a relatively small number of relevant papers. For example, Kamal et al. [28] developed a simple database-supported web interface to assist manual patient search for breast cancer trials. By providing searches according to a fixed set of "common" inclusion/exclusion criteria, it only achieves patient-pruning: since many trials come with their unique criteria, additional manual work is still needed to finally identify the patients.

Lonsdale et al. [29] investigated a more sophisticated system that attempts to convert inclusion/exclusion criteria into standard database queries and use them to retrieve qualified patients. While they have demonstrated some encouraging results on, for example, clinical trials on the effect of different sources of garlic on cholesterol levels, their approach also has some problems. For example, because the patient database is usually constructed independently from and cannot anticipate future clinical trial needs,

many trial criteria cannot be converted easily into database queries. Furthermore, important patient information often exists in free-text documents, which cannot be examined easily by database queries.

For free-text based patient search, Rokach et al. [30] reported some success with a program that uses regular expression match to find patients qualifying for some diabetes trials. In this case, regular expressions are used to summarize and then search for word patterns in patient reports that can confirm the criteria of the trials. Despite its success, regular expression match also has a challenging problem: since the patient text relevant to a given trial criterion may come in diverse forms and patterns, finding a "good" regular expression for them often requires experience and ingenuity; automation is difficult.

In recent years, in the broad field of natural language processing (NLP), there has been considerable interest in the so-called recognizing textural entailment (RTE) problem, e.g., see [33, 34]. In an RTE problem, one is usually presented with two sentences and needs to decide (using a computer algorithm) whether one sentence, usually longer and more complex, called the text, implies the other, usually shorter and simpler, called the hypothesis. A number of techniques have been proposed for RTE, ranging from those based on statistical pattern recognition to those based on traditional AI (logic or machine reasoning), as well as those that combine both. While the RTE work is inspirational and contains ideas potentially useful to our work, there are some differences between its scope and our current problem. For example, since the objective of RTE is domain-independent natural language understanding, current RTE techniques generally try to avoid using application-specific training data and domain knowledge. In our problem, such training data and knowledge are available and play an important part in the search algorithms. Furthermore, RTE is still a relatively new field [with the first TREC (Text Retrieval Conference) contest started around 2006] and much work is still needed to develop truly robust and effective algorithms.

In an interesting study, Li et al. [35] used an natural language processing (NLP) tool called MedLEE to convert free text patient reports to a structured form and then designed queries to search for patients for a stroke related clinical trial. While their NLP based technique seems to work better than a competing technique, the queries were designed manually.

Finally, since databases are widely used to store patient information, it may seem that one could solve the automatic patient search problem by simply adding some new search friendly fields in each patient's database record. For example, one can introduce a new field for "post-menopausal status", with 1 for "Yes" and 0 for "No". In this way, when a clinical trial contains a criterion "the patient must be post-menopausal", the patient search can be done easily by finding all those with a 1 in their "post-menopausal status" field. This approach, although attractive on first look, has a fundamental problem: since we cannot always anticipate future clinical trial criteria, we may not be able to provide fields for them ahead of time. Furthermore, there is also a practical problem: to manually add such fields to current/future patient data records can be very costly or even prohibitive; to do so automatically, on the other hand, requires one to solve the automatic patient search problem in the first place.

3.3 Classification Problem on Free-text medical records

Given a clinical trial, the problem of identifying qualified patients from free-text medical reports can be reduced to a simpler, but equivalent problem: for each trial criterion, find all patients whose reports contain sentences that meet it. In Table 3.1, we have shown three criteria of a trial currently active at Aurora Health Care, a major hospital group in Milwaukee, Wisconsin. Also shown in Table 3.1 are some "positive" sentences from patient reports that meet these criteria. From Table 3.1 we see that for a given criterion, the positive sentences can look very different from the criterion and from each

TABLE 3.1: Trial Criteria and Qualified Sentences

Criterion 1. Primary tumor must be palpable.
1) Within the right breast, an oval dense mass measuring 4 x 4-cm was noted. 2) There is, however, a 3 x 2.5 cm, firm area on the upper inner quadrant of the left breast. 3) The patient has noted a lumpiness in her right upper breast.
Criterion 2. Tumor is ER positive.
1) Estrogen was 75% positive. 2) She understands that histology reveals a poorly differentiated invasive ductal carcinoma that is estrogen and progesterone receptor positive. 3) Estrogen receptor was 100% positive.
Criterion 3. Patient must be post menopausal.
1) Post menopausal woman with new diagnosis of left-breast infiltrating lobular carcinoma. 2) Last menstrual period was 10 years ago. 3) She has had a total abdominal hysterectomy, bilateral salpingo-oophorectomy.

other (this is especially true for criterion 1). Hence, to improve search performance we need to account for such variations. One way to do this is to use a small number of manually found positive examples or sentences associated with a criterion, also known as the training data in patient search.

3.4 Sub-tree Pattern Matching Method

Figure 3.1 illustrates our overall approach to the patient search problem and it consists of three parts. First, for a given clinical trial, the trial criteria and their associated training data are used to produce a set of criterion models, also called search models. Then, these models are used to search for patients whose reports contain text that satisfy the trial criteria. Finally, if so desired, human confirmation of search results are allowed and the search results are used to modify criterion models (a kind of feedback). A very simple model, often used in information retrieval, is a set of keywords, in no

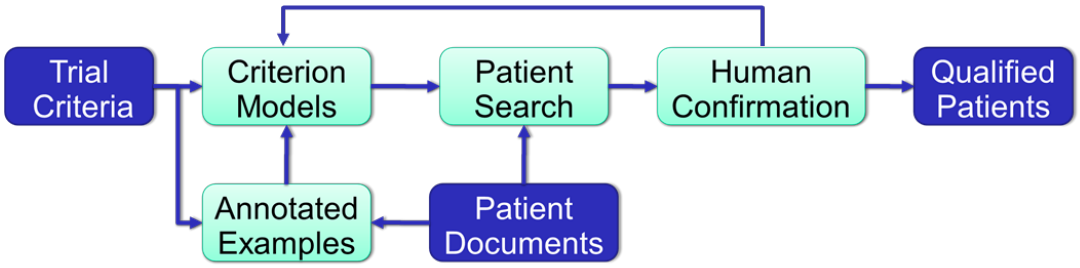


FIGURE 3.1: Illustration of our sub-tree pattern matching approach.

TABLE 3.2: Trial Criteria and Manually Generated Keywords

Criterion 1. Primary tumor must be palpable.
Related keywords: mass, lump, density, prominence, abnormality, thickening, area, finding, palpable, firm, hard, large, dense, irregular, well-defined, note, notice, feel.
Criterion 2. Tumor is ER positive.
Related keywords: estrogen, receptor, ER, positive
Criterion 3. Patient must be post menopausal.
Related keywords: postmenopausal, last menstrual, surgical menopause, salpingo-oophorectomy, (50+)-year-old

particular order, that are deemed important to the criterion and/or commonly seen in the training data associated with the criterion. Table 3.2 shows an example: a criterion and a set of manually selected keywords. Using the keywords as the query, sentences that contain a sufficient number of the keywords are retrieved and deemed positive. The advantage of this model is its simplicity (can be constructed easily by users without much knowledge in computer science or NLP). The disadvantage is that this model is relatively inaccurate because the meaning of a sentence is not only determined by its words but also by how they are organized (i.e., ordered). One way to organize the keywords is through a regular expression [36]. In theory, a regular expression can be used to specify all the possible ways that the keywords are ordered/organized into a sentence that meets the trial criterion. The advantage of regular expression is that it is more accurate and regular expression search can be implemented very efficiently. The disadvantage is that developing a good regular expression generally requires more

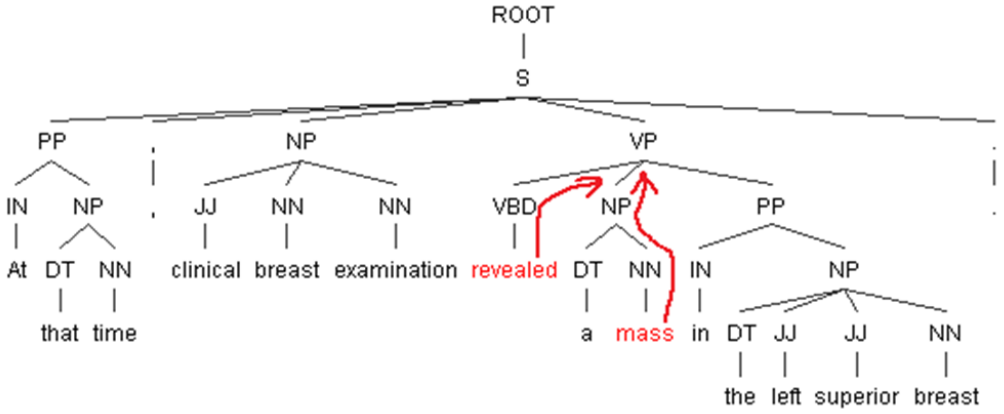
computer science and NLP knowledge therefore, is more difficult for a medical user. Furthermore, the process of developing a regular expression also requires human ingenuity and is difficult to automate.

In this work, we used a search model that organizes keywords into a set of subtrees that capture the grammatical relations between the keywords, thereby to some extent capturing the meanings of various keyword combinations in the positive sentences. Under this model, the problem of searching for positive patient sentences becomes that of searching for keyword subtrees in patient sentences. The subtree model has comparable and often better accuracy than regular expressions but is easier to automate hence, is easier to use for medical workers.

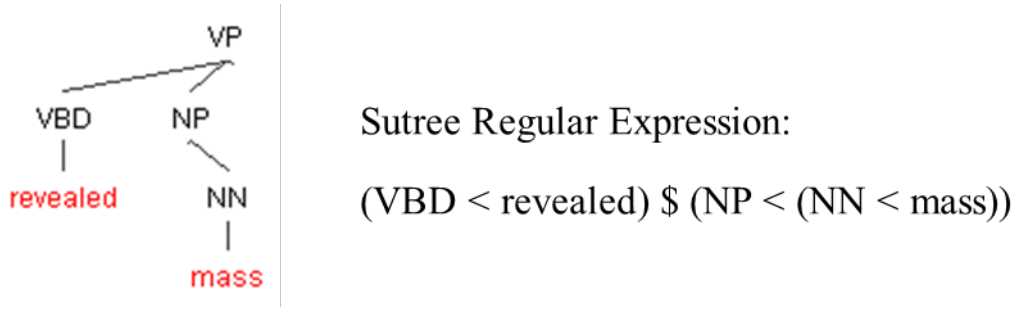
Suppose we have generated a set of keywords, either manually or automatically, from a trial criterion and its associated training sentences. Our subtree model is constructed in three steps. First, each training sentence is parsed by a syntax parser (we used the Stanford parser [37]) into a parse tree which describes sentence's syntax structure; see an example in Figure 3.2 (a). Second, for each parse tree, all leaves that are keywords are located and from them, a program back-tracks up the tree for three levels to generate a set of subtrees; again see Figure 3.2 (b). Finally, all subtrees found in this way are collected and represented as tree regular expressions and these will be used as search models for the given criterion; see Figure 3.2 (b).

We end this subsection with several remarks. First, given the keywords, the subtrees and their representations (tree regular expressions) are produced automatically. Second, given the subtree models for a trial criterion, the process of searching for positive sentences amounts to a tree regular expression match and a number of efficient implementations exist that can be used (we used one from Stanford [37]). Third, for a given criterion, keywords can be generated manually by inspecting the criterion and its training data. This requires a little time but is not too difficult and can be done relatively easily by a medical worker. But to make things even easier, we also developed

techniques for automatic keyword generation.



(a)



(b)

FIGURE 3.2: A parsed tree and subtree (a) syntactic tree of a parsed tree (b) subtree pattern with keywords

3.5 Keyword Generation

3.5.1 Phrase Similarity

Suppose we have a single trial criterion and a set of (training) patient sentences that match the criterion. The problem is to generate a set of keywords from these. Our automatic keyword generation technique starts with a small number of manually selected seed keywords, which are usually words highly relevant to the criterion and occurring frequently in training data. For example, for criterion 1 in Table 3.1

("Primary tumor must be palpable"), we found the words "palpable" and "mass" in the training data as seed keywords (notice that in the training sentences, "mass" is mentioned many times but not "tumor" see Table 3.1).

Since the same idea is often expressed in different words, to find additional keywords it would be reasonable to search in the training data for words similar to the seed keywords. For example, for each seed keyword, we could find other words in the training data that are similar. As a measure of similarity, we experimented with those provided by WordNet [38]. This, however, did not work well. Based on our results, we believe that this may be caused by two factors. First, the WordNet is designed for a general domain (e.g., newspaper articles) not for the breast cancer domain. Thus, two words that are closely related in the latter (e.g., mass and density) may not be so closely related in the former; vice versa. Second, even for words in the general domain, WordNet similarity is not always reliable.

To improve this situation, we experimented with the idea of "using more context", i.e., the intuition that one can make a better decision (on how similar two words are) by looking at more data (the neighbors of these two words). Specifically, we can combine some seed keywords into seed phrases and look for phrases in the training sentence that are similar in meaning but different in words from the seed phrases.

To measure the similarity between the seed phrase and a target phrase, we used a simple matching based algorithm that, in abstract, can be described as follows. Let $\boldsymbol{x} = [x_1, x_2, ..., x_m]$ be a seed phrase of m words and $\boldsymbol{y} = [y_1, y_2, ..., y_n]$ be a target phrase of n words. A match between x and y is an $m \times n$ binary matrix $\boldsymbol{h} = [h_{i,j}]$, where $h_{i,j} = 1$, if x_i is matched to y_j, 0 otherwise. Then we can measure the similarity of $\boldsymbol{x}$ and $\boldsymbol{y}$ by

$$S(x, y) = \max_{\boldsymbol{h}} \sum_{i=1}^{m} s(x_i, h_i \boldsymbol{y}) \tag{3.1}$$

Where h_i is the ith row of $\boldsymbol{h}$, $h_i \boldsymbol{y}$ is the word in $\boldsymbol{y}$ that x_i is matched to, $s(x_i, y_j)$ is the word similarity between x_i and y_j (in this case, the WordNet similarity), and the

TABLE 3.3: Seed Phrase and Automatically Generated Keywords

Criterion 1. Primary tumor must be palpable.
Seed Phrase: palpable mass **Matched Phrase**: irregular mass, palpable density, a large mass, a palpable mass, a palpable lump, hard irregular mass, ill-defined mass, palpable abnormality, well-defined multilobulated mass. **Related keywords**: mass, lump, density, abnormality, area; palpable, hard, large, irregular, well-defined, ill-defined; note, revealed, appeared, identified, noticed, felt, prompted, appreciated, discovered, left.

maximization over $\boldsymbol{h}$ is subject to the unique-matching constraint that each x_i can be matched to only one y_j, i.e.,

$$\sum_{j=1}^{n} h_{i,j} = 1$$

for all i. To perform the maximization, we can use a variety of techniques. Since in our application, the value of m and n are usually very small, for the sake of simplicity, we used exhaustive search.

Once a matching phrase is found for a seed phrase, the words in the phrase that match those in the seed phrase will be taken as keywords. For example, for criterion 1 of Table 3.3, we used the seed phrase "palpable mass" and found a similar phrase "palpable density" and this allows us to add "density" to the list of keywords. Table 3.3 also shows a number of other keywords found this way.

3.5.2 Word Relatedness/Similarity Measured Based On Medical Contexts

Another way to calculate the words relatedness/similarity is to use the medical domain contexts. Intuitively, the idea is that if two words have similar co-occurrences with all other words, they are deemed to be closely related. Suppose we have a set of medical documents which are parsed into the sentences. Suppose that x and y are two words,

and we want to measure their relatedness/similarity. The distance between x and y can be calculated by the divergence:

$$d(x, y) = \sum p(w_i|x) \left(\log p(w_i|x) - \log p(w_i|y)\right) \tag{3.2}$$

, where $p(w_i|x)$ and $p(w_i|y)$ is the conditional probability of an arbitrary word w_i given x and y respectively, and the sum is over all word w_i. The conditional probability $p(w_i|x)$ can be found by

$$p(w_i|x) = \frac{p(w_i, x)}{p(x)} = \frac{\textit{the number of sentences containing both } w_i \textit{ and } x}{\textit{the number of sentences containing } x} \tag{3.3}$$

There are several methods to measure our distance:

1. In practice, the distance measure, specifically, the way to compute $p(w_i|x)$ can be modified. For example, instead of considering all words for w_i, we can remove some words that are useless (e.g., stop words or irrelevant words). Similarly, we can also modify $p(w_i|x)$ by considering only sentences in which w_i and x are within a certain distance (e.g., 5 words a part) or within certain distance in a parse tree.

2. $d(x, y)$ is not symmetrical. For a symmetrical measure, we can use $\frac{d(x,y)+d(y,x)}{2}$.

3. We can impose additional constrains on x and y before measuring their distance. For example, we can require that they have the same part of speech label (e.g., both be noun, both be verb, etc.). Similarly, we can also require that they have the same grammatical roles (we may get this information from parsed trees).

Finally, instead of using word condition probabilities, we can also use word correlations. Suppose we have m sentences in the document and there are total of n different words in the documents (each sentence may contain only a few words, much less than n). Denote each sentence by d_i, $i = 1, 2, ..., m$. Then $d_i = [d_{i1}, ..., d_{in}]^T$, where d_{ij} is either a

TABLE 3.4: Experiments of Word Relatedness

Word	Top 5 Ranking of Related Words
mass	lesion; nodule; area; malignancy; density
hormone	estrogen; endocrine; oestrogen; tamoxifen; progesterone;
menopause	postmenopause; observe; age; baseline; year
biopsy	ultrasound; remove; excision; dissect; mri

word frequency or a binary indicator variable (1 if word j is in d_i, 0 if not). Define matrix $A = [d_1, d_2, ..., d_m]$ and correlation matrix:

$$C = AA^T \tag{3.4}$$

This is the correlation matrix, and each row is a correlation vector. We can normalize each row so that it sums to 1. Then we can compute the directed divergence between two normalized rows as the distance between two words. To measure the word stability in the co-occurrence matrix, we use the entropy:

$$entropy = \sum p_i \log_2 p_i \tag{3.5}$$

We used it to remove the unstable words. To implement this method, we use two sources of data. One is from our 644 free-text Surgeon Visit Notes of 142 patients. The total sentences are 46921. The other one is Breast Cancer Clinical Reports from PubMed [39] in the latest 5 years until 2012. There are 3515 abstracts and total 41982 sentences. Table 3.4 shows an experimental results of the top 5 rankings of most related word. In this experiment, we removed the words with high entropy (> 0.3). Most of these words are the stop words, like "and", "of", "the", "in", "with", "to", "for", "a", etc. And some non-stop words also with high entropy(> 0.3), for example, "patient", "breast", "cancer", "treatment", "tumor", "no", "group", "study", "receive", "clinic", "report", "woman", "compare", "result", "dose", "therapy", "chemotherapy". For the

measure of the words relatedness, these high entropy words are filtered out.

3.6 Bernoulli Model and Bayes Ratio Classification

In section 3.4, we described the subtree based technique for patient search [40]. In this approach, subtree models were derived from training patient text data and later used to identify qualified patients from patient reports. While this approach produced some promising results, it is somewhat complex and require manually selected seed words. In order to improve this, we propose a new sentence classification method in this section. Our new approach uses a trained Bernoulli Model which is created from our training data (qualified or unqualified sentences) [41]. The trained Bernoulli model calculates the probability of a new word sequence (test sentence) to make a classification decision. Our method has several advantages. There is no required special pre-processing, such as syntactic parsing, POS tagging etc. This is beneficial for the free form text medical report which may contain grammatical mistakes during transcription. The algorithm is fast, scalable, and could be easily applied to any natural language. The model's decisions are based on a specific likelihood threshold, which can change/optimize the recall and precision.

We used Bernoulli models to determine if sentences in a patient's medical report meets a clinical trial criterion. Bernoulli models and other language models were originally used for information retrieval (IR) [42]. Indeed, some previous work [43, 44] indicate that when retrieving sentences, which are usually much shorter than typical documents, the Bernoulli models perform better than other popular language models, such as multinomial (or bag of words) models. Despite some similarities to the IR models, our models differ from them in that in IR, a model is estimated for each document or sentence, while in our work, a model is estimated for a class of sentences. Figure 3.3 illustrates our overall approach to the patient search problem by using Bernoulli Model.

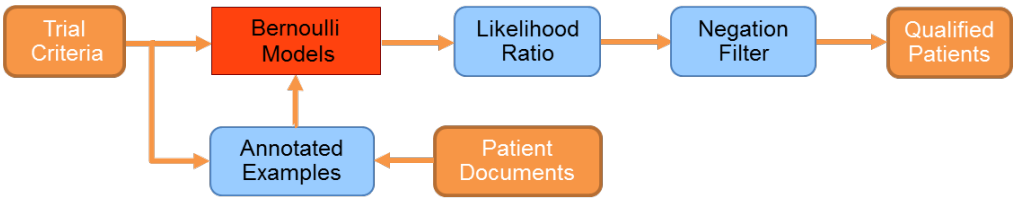


FIGURE 3.3: Illustration of overall approach by using Bernoulli model

For training the model, firstly, we identified a group of patients that associated to one of the criterion. Within the reports of these patients, we manually collected the sentences that would be considered consistent with the criterion. These sentences are labeled as "positive" examples, while the remaining sentences are labeled as "negative". Both sets of sentences are used to train the Bernoulli Model. Then, we use the trained Bernoulli Model to search for qualified patients whose reports contain text that satisfy the criterion. Finally, the results of qualified patients can be found after Negation detection filtering is performed.

Within the Bernoulli Model, we use the binary Bernoulli vector to represent the vocabulary which is collected from the training datasets. Each term in the binary vector represents a word: the 1 indicates presence of the term in the vector model, 0 otherwise. The single-word Bernoulli vector is defined as $\vec{\boldsymbol{x}} = [x_1, x_2, ..., x_m]$, where x_i is a binary value to indicate if the word w_i appears or not in a given sentence. Let the sentence "s" be represented by the single-word Bernoulli vector.

$$p(\mathbf{s}) = \prod_{i=1}^{|\vec{\boldsymbol{x}}|} p(w_i)^{x_i}(1 - p(w_i))^{1-x_i} \tag{3.6}$$

Where $p(w_i)$ is the probability of word w_i in the training data set. The sentence probability is the product of each word occurring both in and not in the sentence. Because the Bernoulli Model uses the word occurrence rather than the term frequency of the words, the probability calculation of the word $p(w_i)$ is:

$$p(w_i) = \frac{number\ of\ sentences\ containing\ w_i}{total\ number\ of\ sentences} \tag{3.7}$$

Similarly, the two-words Bernoulli vector can be defined as: $\vec{\boldsymbol{x}} = [< x_i, x_j >, \ldots]$, where $i \neq j$ and $< x_i, x_j >$ is a binary value (0 or 1) to indicate if the two words $< w_i, w_j >$ appear or not in a given sentence. For the two-class training case, we need two different training datasets. One set is constructed by criterion qualified (positive) sentences, the other is constructed by the irrelevant (negative) sentences. The Bernoulli Model is trained from both the positive and negative datasets. We will use H_1 to indicate the positive class and H_0 to indicate the negative class. The probability of word w_i from a positive dataset is defined as $p(w_i|H_1)$, and from a negative dataset is defined as $p(w_i|H_0)$.

Within the two-class Bernoulli Model, there will be a problem with the estimation of probability of the unseen words. Instead of assigning zero probability to the unseen words in the positive or negative training datasets, we will smooth the model by giving a very small probability value ϵ to unseen words. The sentence represented by the Bernoulli vector for the positive and negative class was defined as

$$p(\boldsymbol{s}|H_1) = \prod_{i=1}^{|\vec{\boldsymbol{x}}|} p(w_i|H_1)^{x_i}(1 - p(w_i|H_1))^{1-x_i} \tag{3.8}$$

$$p(\boldsymbol{s}|H_0) = \prod_{i=1}^{|\vec{\boldsymbol{x}}|} p(w_i|H_0)^{x_i}(1 - p(w_i|H_0))^{1-x_i} \tag{3.9}$$

The sentences are ranked by classification ratio which is using the Bayes' Rule:

$$Ratio(H_1|\boldsymbol{s}) = \frac{p(H_1|\boldsymbol{s})}{p(H_0|\boldsymbol{s})} \propto \frac{p(\boldsymbol{s}|H_1)}{p(\boldsymbol{s}|H_0)} \tag{3.10}$$

3.7 Naive Bayes Model

The Bernoulli Model differentiates itself from Naive Bayes by the word probability estimation methods and the classification rules. The Naive Bayes Model is based on

n-grams probabilistic learning method. Due to sparse training examples, the unigram model and bigrams model (with add-one smoothing) were used. The unigram $p(w_i)$ is the probability of w_i in the training dataset, defined as

$$p(w_i) = \frac{c(w_i)}{N} \tag{3.11}$$

where $c(w_i)$ is the number of w_i in the training sentences, and N is the total number of words in the training sentences. The bigrams $p(w_i|w_{i-1})$ is the word probability conditioned on its previous word, defined as

$$p(w_i|w_{i-1}) = \frac{c(w_{i-1}w_i)}{\sum_{w_i} c(w_{i-1}w_i)} \tag{3.12}$$

Add-one smoothing in bigrams is defined as

$$p(w_i|w_{i-1}) = \frac{1 + c(w_{i-1}w_i)}{\sum_{w_i} (1 + c(w_{i-1}w_i))} \tag{3.13}$$

Using the unigram Naive Bayes Model, the sentence probability is the product of each word occurring in the sentence $\mathbf{s}$.

$$p(\mathbf{s}) = \prod_{i=1}^{|\mathbf{s}_i|} p(w_i) \tag{3.14}$$

So, the sentence probability based on bigrams Naive Bayes Model is

$$p(\mathbf{s}) = \prod_{i=1}^{|\mathbf{s}_i|} p(w_i|w_{i-1}) \tag{3.15}$$

3.8 Experimental Results

For the trial criteria, we used the 3 criteria in Table 3.1. This selection was based on three considerations. First, the trial criteria came from a trial that is currently active at Aurora (ID: ACOSOGZ1031). Second, they are diverse and representative: criteria 1 and 3 are more difficult to search for since their positive sentences have greater variation while criterion 2 is easier. Finally, using many more criteria would be beneficial but may not be necessary: the 3 criteria used are quite representative of the various levels of difficulties.

For patient data, we used the medical reports of 142 breast cancer patients from Aurora. For each patient, we used only one report, Visit Note Surgery 1 (VNS1), which contains information relevant to the 3 selected trial criteria (criteria 1 to 3 of Table 3.1). The 142 VNS1 reports were parsed into single sentences by an NLP software, resulting in a total of 18,873 sentences. These sentences were then annotated with respect to each of the trial criteria as either "meeting" or "not meeting" the criterion. Based on the annotations, training data (positive sentences) were taken randomly out of all positive sentences for each criterion. The number of training positive sentences are 46, 25, and 45, respectively, for criteria 1, 2, and 3, and the number of training negative sentences were randomly selected 1000, 5000, 5000 for each criterion. Finally, after the training data were taken, the rest of the data were used as testing data. Furthermore, in the testing data for criterion 1, we removed those in the sections for ultrasound and mammogram results since these sections generally do not contain information related to criterion 1. This resulted in 1710, 13,778, and 13,725 testing sentences which includes 126, 70 and 103 positive sentences respectively for criteria 1, 2 and 3. The experimental results are shown in Table 3.5 in terms of precision, recall and F-score.

The three criteria for our experiment represent different characteristics of searching complications. Our results (see Table 3.5) indicate that the subtree technique provides

significantly better performance than all other techniques (e.g., see the F-scores) except for the regular expression where the performance is competitive. As described in Section 3.4, the semi-auto subtree technique is preferable because using parse trees the search models for this technique can be generated automatically. And we also proposed an auto method based on Bernoulli Model in Section 3.6 to deal with different kinds of searching difficulties while maintaining good results.

There are a few of other ways to build the Bernoulli vector. One way is to manually select some keywords for the given criterion, and the Bernoulli vector will be constructed only by those keywords. Our manually selected keywords for each criteria is given in Table 3.2, the same as in our previous paper([40]), and the results are given in the Table 3.3. One other way to build the Bernoulli vector is to use the Unified Medical Language System (UMLS)'s Concept Unique Identifier(CUI) or Semantic Types(ST). We converted each word of our free-text datasets to the UMLS CUI or ST formats.

3.8.1 Previous Results

The results of experiments (Table 3.5[1]) show that the "Manual-word Bernoulli Model" has the best F-score for Criterion 2 and 3, but not for Criterion 1. Comparing with the keywords based methods(Regular Expression and subtree model) by using the same keywords in the same datasets, the "Manual-words Bernoulli Model" has the best results and is easier to implement. However, for the more complicated criteria, such as Criterion 1, which has many variations of sentences, finding the best manual keywords is challenging and tedious work.

For comparison, we also included the results of regular expression, concept search, vector space method(VSM), and LSI. Among these, the regular expressions are manually designed, the concept search works by "normalizing" search keywords and words in patient sentences into UMLS concepts [45] and then perform matching, and the vector

[1]RegEx is Regular Expression; VSM is Vector Space Method; and LSI is Latent Semantic Indexing.

space method (VSM) and Latent Semantic Indexing (LSI) are two commonly used information retrieval methods (see [42, 46]; here we use them to retrieval sentences).

TABLE 3.5: Previous Results on Patient Data

Techniques	Precision, Recall, F-score		
	Criterion 1	**Criterion 2**	**Criterion 3**
Bernoulli(single)	66%, 74%, 0.70	79%, 71%, 0.75	81%, 66%, 0.73
Naive Bayes(single)	59%, 69%, 0.63	80%, 63%, 0.71	82%, 74%, 0.78
Bernoulli(pairs)	53%, 74%, 0.62	36%, 50%, 0.42	66%, 40%, 0.50
Bernoulli(CUI)	66%, 69%, 0.67	82%, 73%, 0.77	58%, 51%, 0.55
Bernoulli(ST)	51%, 60%, 0.55	57%, 79%, 0.66	19%, 34%, 0.24
Bernoulli(keywords)	55%, 71%, 0.62	96%, 98%, 0.97	96%, 93%, 0.94
Subtree (semi-auto)	90%, 49%, 0.63	89%, 83%, 0.86	73%, 86%, 0.79
RegEx (semi-auto)	37%, 72%, 0.49	84%, 91%, 0.87	67%, 88%, 0.76
Concept Matching	13%, 91%, 0.23	24%, 86%, 0.37	36%, 100%, 0.53
VSM	26%, 37%, 0.31	30%, 65%, 0.42	26%, 100%, 0.41
LSI	42%, 42%, 0.42	17%, 57%, 0.26	20%, 100%, 0.33

3.8.2 Updated Results

In addition to the Bernoulli model, we also applied some of the leading supervised machine learning techniques to our experiments on the data set for the three criteria. These techniques include linear classifiers, such as the logistic regression, stochastic gradient descent(SGD) classifier, and support vector machines, and ensemble classifiers, such as the random forest, gradient boosting, and XGBoost. Furthermore, we tested the bagging and stacking techniques which combine individual classifiers in different ways. First, bagging is used to average the predication of all individual classifiers with the same or different weights. Second, unlike bagging and boosting method, the basic idea of stacking is that the output of one classifier can be used as training data for another classifier. The results are shown in Figure 3.4,3.5, 3.6. Each precision-recall curve presents the performance of each individual classifier, and the area under the curve (AUC) can quantitatively measure the classifier's efficacy. For better comparison, the

AUC of each classifier from the figures are also summarized in Table 3.6. From these results, we can see that an individual classifier may not always perform the best over all criteria but bagging can lead to a classifier that is the best. The Stacking method which combines only three classifiers: logistic regression, SGD classifier and Bernoulli, also achieves good result. The advantage of stacking is that it is fast and scalable; it uses simple and weak classifiers but, after stacking, it is able to achieve similar results to the Bagging method.

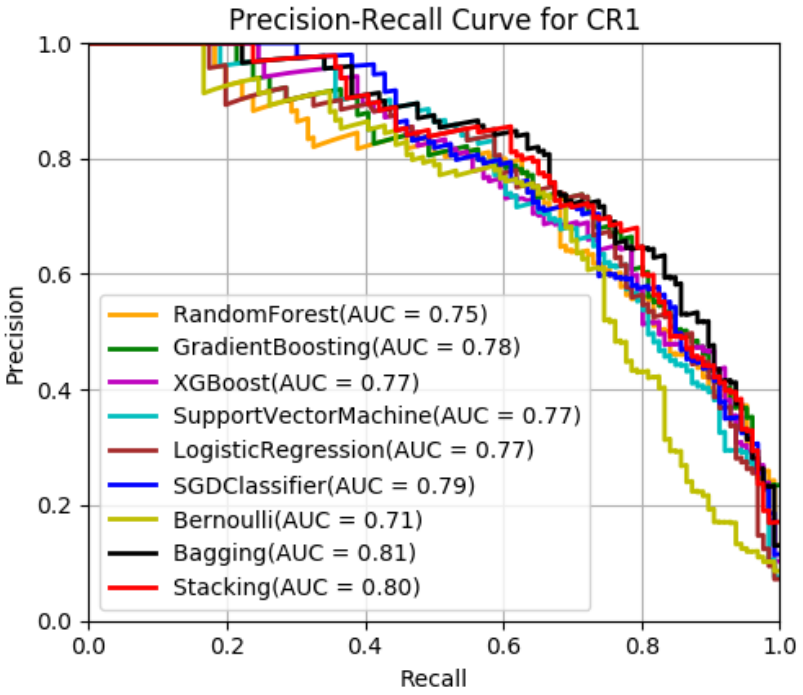


FIGURE 3.4: Precision-recall curve for criterion 1

3.9 Unsupervised Learning of Word Embeddings and Future Work

When we apply machine learning techniques to our sentence classification problem, the most important step is to convert variable length texts/sentences into fixed-length vectors.

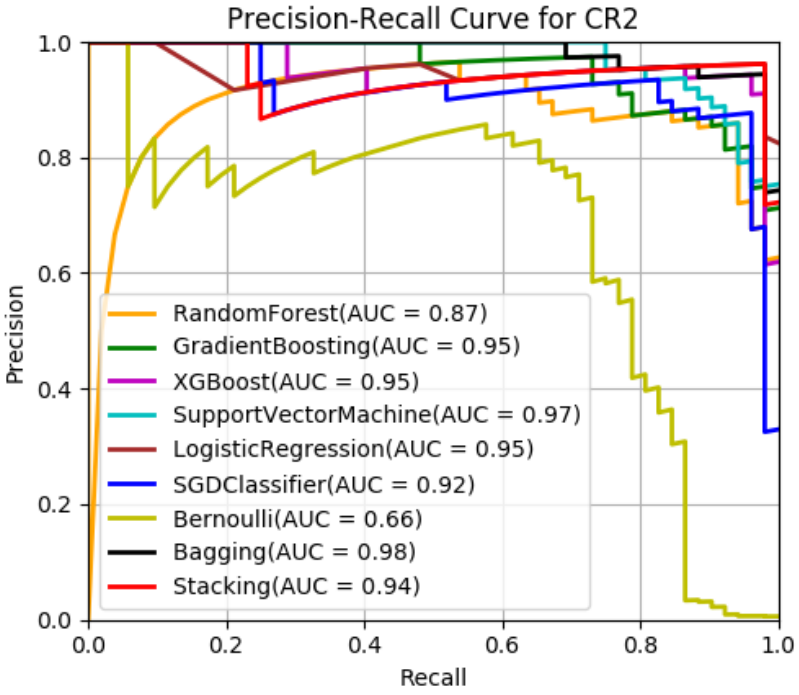


FIGURE 3.5: Precision-recall curve for criterion 2

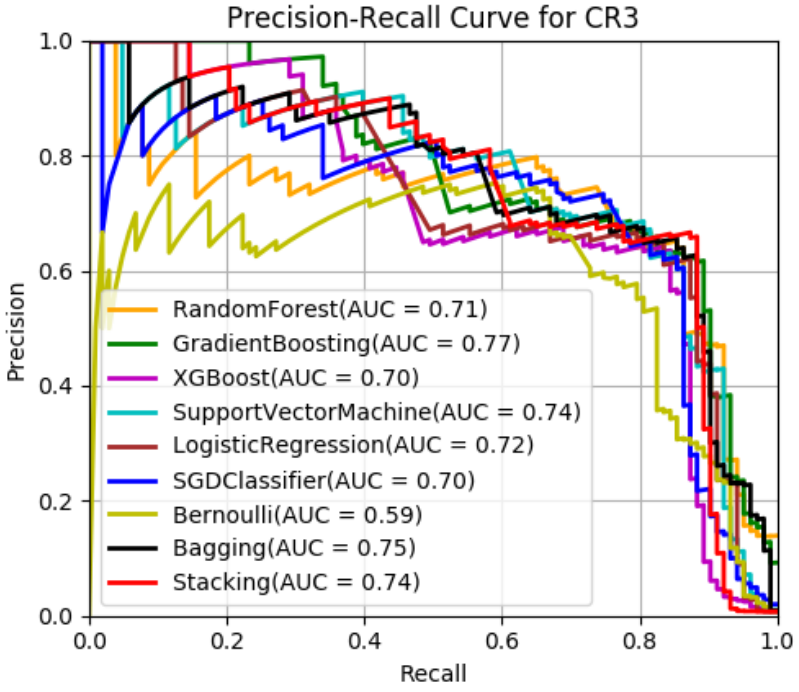


FIGURE 3.6: Precision-recall curve for criterion 3

The vector representation of a sentence needs two considerations: how to convert words into numerical vectors and how to represent the order of word sequences. The challenge is that a sentence usually has different word representations with different length, and the output needs to be a fixed-length representation. Recent advances in natural language processing have provided unsupervised learning algorithms to produce vector representations for texts such as words, sentences, paragraphs or entire documents. The basic idea is to learn vector representation models from huge amounts of unstructured text data in an unsupervised process. Mikovlov et al. [47, 48] proposed a Bag-of-Words model (CBOW) and a Skip-Gram model for learning word embeddings from free-text data. Their word embedding models outperformed other bag-of-words models and techniques for text representation and achieved state-of-the-art results on some text classification and sentiment analysis tasks. Their work can be potentially extended to our sentence classification work. Our initial experiments were using the gensim's doc2vec[49] implementation which is based on the unsupervised learning of word embeddings (Mikovlov et al.). The model was trained on the dump of wiki's latest article data [50]. The size of data is about 13GB. The training time was about 3days and 5 hours for both of the dbow (distributed bag of words) and dm (distributed memory) model by using a typical PC with Intel Xeon X5570 2.93GHZ 16 CPUs and 43 GB RAM. The size of the trained model was about 5.2GB. Then, the trained models were used to inference the vector representation for each new sentence obtained from our breast cancer medical reports. Based on our ground truth and analysis, the trained model didn't make good predications for our medical sentences. The reason might be that the wiki data is too different from medical data. In future work, we plan to re-train the models using data sets from medical domain, such as breast cancer patient medical reports and journals or abstracts from PubMed.gov.

TABLE 3.6: Updated Results on Patient Data

Techniques	**AUC(Area Under Curve)**		
	Criterion 1	**Criterion 2**	**Criterion 3**
RandomForest	0.75	0.87	0.71
GradientBoosting	0.78	0.95	0.77
XGBoost	0.77	0.95	0.70
SVM(kernel)	0.77	0.97	0.74
LogisticRegression	0.77	0.95	0.72
SGDClassifier	0.79	0.92	0.70
BernoulliBayes	0.71	0.66	0.59
Bagging	0.81	0.98	0.75
Stacking	0.80	0.94	0.74

Chapter 4

Maximum A Posteriori Expectation Maximization Clustering Method

In this section, we introduce our unsupervised clustering techniques of Maximum A Posteriori(MAP) Expectation-Maximization(EM) method.

4.1 Clustering Problem

Clustering is to solve the problem of finding the number of components and the parameters of each component. There are many clustering methods, for instance, K-means, hierarchical clustering, density-based clustering, etc. [1, 2, 3, 4, 5, 7, 8]. However, there are two kinds of drawbacks in these traditional clustering methods. First, most of them need to specify the number of components or a threshold/classifier before the clustering algorithm. Second, the local maximum happens in all of the clustering methods, and appears more when the variables dimension increases.

4.2 Motivations

The purpose of our work is to develop a unsupervised clustering method to automatically determine the number of components and the parameters for each component. Meanwhile, the prior knowledge could be used in the clustering process to avoid/improve the local maximum. The resulting algorithm could be useful in the following respects. First, in practice, we can use it without knowing the number of components/mixtures in the data. Second, given some prior knowledge of the data, we can use it to avoid/improve the local maximum during the clustering process. Third, the method can be used in different data types (continuous or binary) with its corresponding distribution models. Finally, the proposed method can be applied to many different areas, such as image processing, data mining, or natural language processing, etc.

4.3 A Short Review of EM Algorithms

In brief review of EM algorithm [13, 14, 15], we derived it with Gaussian Mixture Model. And it also can be extended into another statistical model, for example, the Bernoulli Model. Bernoulli Model is defined for the binary variables, which is useful for language/text processing.

4.3.1 Complete Data and Incomplete Data

First, we need to define the complete and incomplete data. For the complete data, we use x and $x = \{y, z\}$, where y is the observation data, and z is the indicator that which class the y belongs to. For the incomplete data, when the z is unknown, and only y is remained.

For Gaussian Mixture Model, each x_i came from Gaussian distribution $N(\mu_j, \Sigma_j)$ with weight/proportion of π_j, so

$$p(x_i|\boldsymbol{\theta}) = \sum_{j=1}^{K} \pi_j p\left(x_i|\mu_j, \Sigma_j\right) \tag{4.1}$$

, where $\boldsymbol{\theta} = \{\mu_1, \mu_2, ..., \mu_K; \Sigma_1, \Sigma_2, ..., \Sigma_K; \pi_1, \pi_2, ..., \pi_K\}$ and $\sum \pi_j = 1$, $\pi_j > 0$. The probability density of the multivariate normal distribution in d dimension is

$$p(x|\mu, \Sigma) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left\{-\frac{1}{2}(x-\mu)^T(\Sigma)^{-1}(x-\mu)\right\} \tag{4.2}$$

, where $x \in \mathbb{R}^n$, $\mu \in \mathbb{R}^n$, $\Sigma \in \mathbb{R}^{n\times n}$.

4.3.2 Estimation (E) Step

In E step, we calculate the conditional expectation of the **log-MLE** (Maximum Likelihood Estimation) of the complete data.

$$\begin{aligned}
Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) &= E[\log p(x|\boldsymbol{\theta})|y, \boldsymbol{\theta}^{(t)}] \\
&= \sum_i \sum_j E[z_{i,j}|y_i, \boldsymbol{\theta}^{(t)}] \log p(y_i|z_j, \boldsymbol{\theta}^{(t)}) + \sum_i \sum_j E[z_{i,j}|y_i, \boldsymbol{\theta}^{(t)}] \log \pi_j \\
&= \sum_i \sum_j E[z_{i,j}|y_i, \boldsymbol{\theta}^{(t)}] \log \left(\frac{1}{(2\pi)^{d/2}|\Sigma_j|^{1/2}} \exp\left\{-\frac{1}{2}(y_i-\mu_j)^T(\Sigma_j)^{-1}(y_i-\mu_j)\right\}\right) \\
&+ \sum_i \sum_j E[z_{i,j}|y_i, \boldsymbol{\theta}^{(t)}] \log \pi_j
\end{aligned} \tag{4.3}$$

$$\begin{aligned}
E[z_{i,k}|y_i, \boldsymbol{\theta}^{(t)}] &= p(z_{i,k}|y_i, \boldsymbol{\theta}^{(t)}) \\
&= p(z_{i,k} = 1|y_i, \boldsymbol{\theta}^{(t)}) \\
&= \frac{p(y_i|z_{i,k} = 1, \boldsymbol{\theta}^{(t)}) p(z_{i,k} = 1|\boldsymbol{\theta}^{(t)})}{\sum_{j=1}^{K} p(y_i|z_{i,j} = 1, \boldsymbol{\theta}^{(t)}) p(z_{i,j} = 1|\boldsymbol{\theta}^{(t)})}
\end{aligned} \tag{4.4}$$

4.3.3 Maximization (M) Step

In M step, we re-estimate the parameter $\boldsymbol{\theta}^{(t+1)} = \arg\max_{\boldsymbol{\theta}} \left\{ Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) \right\}$ by

$$\frac{\partial Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)})}{\partial \boldsymbol{\theta}} = 0$$

For simplicity, the derivations are in APPENDIX A.

After taking the partial derivative of the **log-MLE** function with respect to $\{\mu, \Sigma, \pi\}$, each parameter is re-estimated as:

$$\hat{\mu}_j^{(t+1)} = \frac{\sum_{i=1}^{n} E[z_{i,j}|y_i, \boldsymbol{\theta}^{(t)}] y_i}{\sum_{i=1}^{n} E[z_{i,j}|y_i, \boldsymbol{\theta}^{(t)}]} \tag{4.5}$$

$$\hat{\Sigma}_j^{(t+1)} = \frac{\sum_{i=1}^{n} E[z_{i,j}|y_i, \boldsymbol{\theta}^{(t)}] (y_i - \hat{\mu}_j^{(t+1)})^T (y_i - \hat{\mu}_j^{(t+1)})}{\sum_{i=1}^{n} E[z_{i,j}|y_i, \boldsymbol{\theta}^{(t)}]} \tag{4.6}$$

$$\hat{\pi}_j^{(t+1)} = \frac{\sum_{i=1}^{n} E[z_{i,j}|y_i, \boldsymbol{\theta}^{(t)}]}{n} \tag{4.7}$$

4.4 MAP EM Algorithms

In Bayesian theory, if the posterior distribution is in the same distribution family as the prior probability distribution, they are defined as conjugate distributions. And the prior is called the conjugate prior.

4.4.1 Normal-Inverse-Wishart Prior

The conjugate prior [17] of multivariate normal(Gaussian) distribution is the Normal-inverse-Wishart distribution. The notation is $(\mu, \Sigma) \sim \text{NIW}(m_0, \lambda, S_0, \nu_0)$. The

probability density function can be written as:

$$p(\mu, \Sigma | m_0, \beta_0, S_0, \nu_0) = N\left(\mu | m_0, \frac{1}{\beta_0}\Sigma\right) W^{-1}(\Sigma | S_0, \nu_0) \tag{4.8}$$

The first term in Eq. 4.8 is the multivariate normal distribution in d dimension as:

$$N\left(\mu | m_0, \frac{1}{\beta_0}\Sigma\right) = \frac{1}{(2\pi)^{d/2}\left|\frac{1}{\beta_0}\Sigma\right|^{1/2}} \exp\left\{-\frac{1}{2}(\mu - m_0)^T \left(\frac{1}{\beta_0}\Sigma\right)^{-1}(\mu - m_0)\right\} \tag{4.9}$$

The second term in Eq. 4.8 is the probability density function of the Inverse-Wishart, which is written as:

$$W^{-1}(\Sigma | S_0, \nu_0) = \frac{|S_0|^{\nu_0/2}}{2^{\frac{\nu_0 d}{2}}\Gamma_d(\frac{\nu_0}{2})} |\Sigma_j|^{-(\nu_0+d+1)/2} \exp\left\{-\frac{1}{2}Tr(S_0 {\Sigma_j}^{-1})\right\} \tag{4.10}$$

, where Σ and S_0 are positive definite matrices, v_0 is degrees of freedom, Γ_d is the multivariate Gamma function, and $Tr(\cdot)$ is the trace of the matrix.
The parameters m_0, β_0, S_0, ν_0 have following interpretations: m_0 is the prior means of μ; β_0 represents how the prior m_0 is closed to the true means of μ; S_0 is the prior mean of Σ; ν_0 shows how the prior S_0 is closed to the true means of Σ.

4.4.2 Dirichlet Prior

Dirichlet prior is used on the proportions/weights (π) of the mixture, denoted as $\pi \sim Dir(\alpha)$. The probability function is written as:

$$Dir(\boldsymbol{\pi} | \boldsymbol{\alpha}) = \frac{1}{B(\boldsymbol{\alpha})} \sum_{i=1}^{K} {\pi_i}^{\alpha_i - 1} \tag{4.11}$$

The beta function is written as

$$B(\boldsymbol{\alpha}) = \frac{\prod_{i=1}^{K} \Gamma(\alpha_i)}{\Gamma(\sum_{i=1}^{K} \alpha_i)} \tag{4.12}$$

, where $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, ..., \alpha_K)$.

4.4.3 MAP Estimation

Now, we will introduce the MAP EM [4, 5] method, which has the same E step as the general EM algorithm, but the M step needs to be changed. After adding the priors on $\boldsymbol{\theta} = \{\boldsymbol{\mu}, \boldsymbol{\Sigma}\}$ and $\boldsymbol{\pi}$, the conditional expectation of the **log-MLE** (Maximum Likelihood Estimation) of the complete data can be re-written as:

$$\begin{aligned} Q'(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) = \sum_i \sum_j E[z_{i,j}|y_i, \boldsymbol{\theta}^{(t)}] \log p(y_i|z_j, \boldsymbol{\theta}^{(t)}) + \sum_i \sum_j E[z_{i,j}|y_i, \boldsymbol{\theta}^{(t)}] \log \pi_j \\ + \sum_j p(\pi_j|\alpha) + \sum_j \log p(\boldsymbol{\theta}^{(t)}|m_0, \beta_0, S_0, \nu_0) \end{aligned} \tag{4.13}$$

We will write the completed form by inserting the Eq. 4.8 and Eq. 4.11.

For a clean form, we let $r_{i,j} = E[z_{i,j}|y_i, \boldsymbol{\theta}^{(t)}]$. So,

$$\begin{aligned} Q'(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) = \sum_i \sum_j r_{i,j} \log \left(\frac{1}{(2\pi)^{d/2}|\Sigma_j|^{1/2}} \exp\left\{ -\frac{1}{2}(y_i - \mu_j)^T (\Sigma_j)^{-1} (y_i - \mu_j) \right\} \right) \\ + \sum_i \sum_j r_{i,j} \log \pi_j + \sum_j \log \left(\frac{1}{B(\boldsymbol{\alpha})} \right) + \sum_j (\alpha_i - 1) \log \pi_j \\ + \sum_j \log \left(\frac{1}{(2\pi)^{d/2} \left| \frac{1}{\beta_0} \Sigma_j \right|^{1/2}} \exp\left\{ -\frac{1}{2}(\mu_j - m_0)^T \left(\frac{1}{\beta_0} \Sigma_j \right)^{-1} (\mu_j - m_0) \right\} \right) \\ + \sum_j \log \left(\frac{|S_0|^{\nu_0/2}}{2^{\frac{\nu_0 d}{2}} \Gamma_d(\frac{\nu_0}{2})} |\Sigma_j|^{-(\nu_0+d+1)/2} \exp\left\{ -\frac{1}{2} Tr(S_0 {\Sigma_j}^{-1}) \right\} \right) \end{aligned} \tag{4.14}$$

Then, we re-estimate the parameters $\boldsymbol{\theta}^{(t+1)} = \arg\max_{\boldsymbol{\theta}} \left\{ Q'(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) \right\}$ by

$$\frac{\partial Q'(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)})}{\partial \boldsymbol{\theta}} = 0$$

For simplicity, the derivations are in APPENDIX B. So each parameter is re-estimated as:

$$\hat{\mu}_j^{(t+1)} = \frac{\sum_{i=1}^{n} r_{i,j} y_i + \beta_0 m_0}{\sum_{i=1}^{n} r_{i,j} + \beta_0} \tag{4.15}$$

$$\hat{\Sigma}_j^{(t+1)} = \frac{S_0 + \sum_{i=1}^{n} r_{i,j} \left(y_i - \hat{\mu}_j^{(t+1)} \right)^T \left(y_i - \hat{\mu}_j^{(t+1)} \right) + \beta_0 \left(\hat{\mu}_j^{(t+1)} - m_0 \right)^T \left(\hat{\mu}_j^{(t+1)} - m_0 \right)}{\sum_{i=1}^{n} r_{i,j} + \beta_0 + \nu_0 + D + 2} \tag{4.16}$$

$$\hat{\pi}_j^{(t+1)} = \frac{\sum_{i=1}^{n} r_{i,j} + \alpha_j - 1}{n + \sum_{j=1}^{K} \alpha_j - K} \tag{4.17}$$

Compared with the parameters from General EM in Eq.4.5, Eq.4.6 Eq.4.7, the parameters from MAP EM in Eq.4.15, Eq.4.16 Eq.4.17 are different by including the priors' parameters of $\boldsymbol{\alpha}, \boldsymbol{m_0}, \boldsymbol{\beta_0}, \boldsymbol{S_0}, \boldsymbol{\nu_0}$. These priors' parameters in MAP EM algorithm can be used to avoid/improve the local maximum, which happens in General EM algorithm.

4.4.4 Convergence

We have two convergence criteria to measure if the iteration of MAP EM algorithm is need to stop. The first stopping criterion is if the maximum number of iteration (for example $max\_iter = 5000$) is reached. The second stopping criterion is if the error between successive iterations is smaller than a certain value ϵ (for example $\epsilon = 10^{-10}$). The error can be calculated as:

$$Err\left(\boldsymbol{\theta}^{(t+1)}, \boldsymbol{\theta}^{(t)}\right) = \frac{|Q(\boldsymbol{\theta}^{(t+1)}) - Q(\boldsymbol{\theta}^{(t)})|}{|Q(\boldsymbol{\theta}^{(t)})|} \tag{4.18}$$

We can use the gradient of the Error function to evaluate the convergence conditions.

$$\nabla_{\boldsymbol{\theta}} Err\left(\boldsymbol{\theta}^{(t+1)}, \boldsymbol{\theta}^{(t)}\right) = const * (|\boldsymbol{\theta}^{(t+1)} - \boldsymbol{\theta}^{(t)}|) \tag{4.19}$$

Then, the convergence condition can be simplified as:

$$\frac{|\boldsymbol{\theta}^{(t+1)} - \boldsymbol{\theta}^{(t)}|}{|\boldsymbol{\theta}^{(t)}|} < \epsilon \tag{4.20}$$

4.5 Synthetic Result

In order to show how MAP EM algorithm works, a synthetic data is created to test and evaluate the performance of MAP EM algorithm. The synthetic data is 2D Gaussian Mixture Data with different proportions, as shown in Figure 4.1. The MAP EM starts from the initials by using K-means with K=10, shown in Figure 4.2 (a), the convergence result is shown in Figure 4.2 (b). By comparing with ground truth, we can conclude if the MAP EM gets the correct result. For comparisons, it is also tested with some other traditional clustering methods. Figure 4.3 shows the result by using the general EM algorithm, which can be seen as MAP EM without any prior information. The general EM starts with the same initials as the MAP EM method, but it converged into the local maxima with the wrong number of components. However, even if we use the correct number of clusters (K=3), the result (in Figure 4.4) is still wrong by using K-means method. The result from non-parametric method: Chinese Restaurant Process(CRP)[17, 18], shown in Figure 4.5, could be used for clustering correctly, but the problem is that it is slow at expense of extra computational time of Gibbs Sampling and many iterations to get the statistical clustering result.

4.6 Summary

Limitations of General EM

- Predefined number of components or threshold/classifier
- Choice of the initial estimates
- Local maximum

Advantages of MAP EM:

- Avoid/improve local maximum by adding prior information
- Automatically determine the number of components
- Faster than Non-parametric methods
- Converged, deterministic

In the next chapter, our proposed MAP EM method with the Gaussian Mixture Model is used to automatically detect the objects in 3D CT luggage image. Our goal is to find the proper number of objects and segment each object in the 3D CT luggage image.

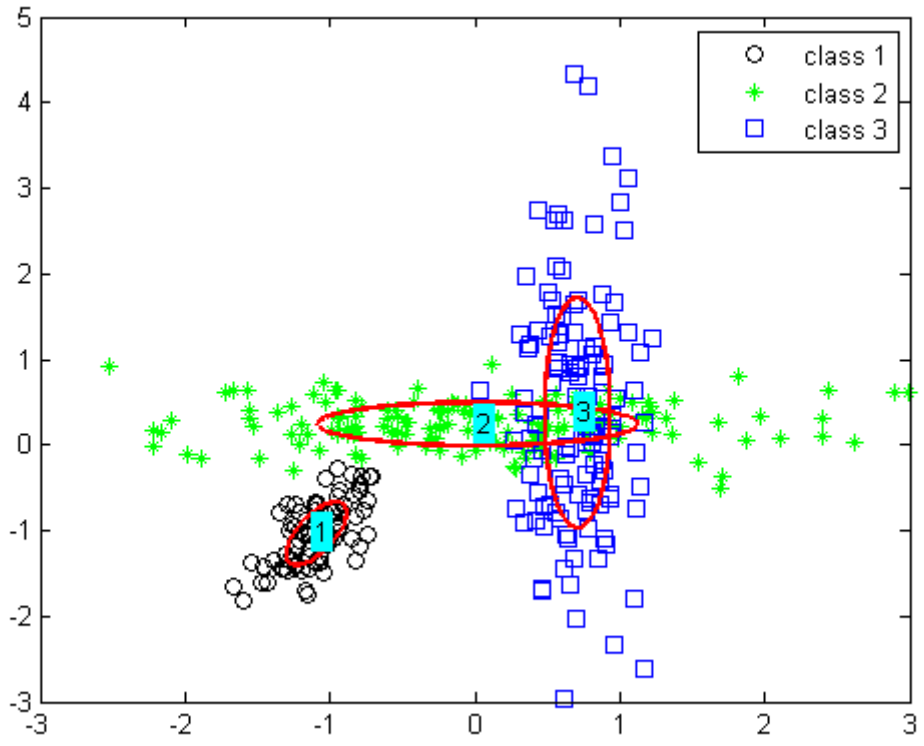
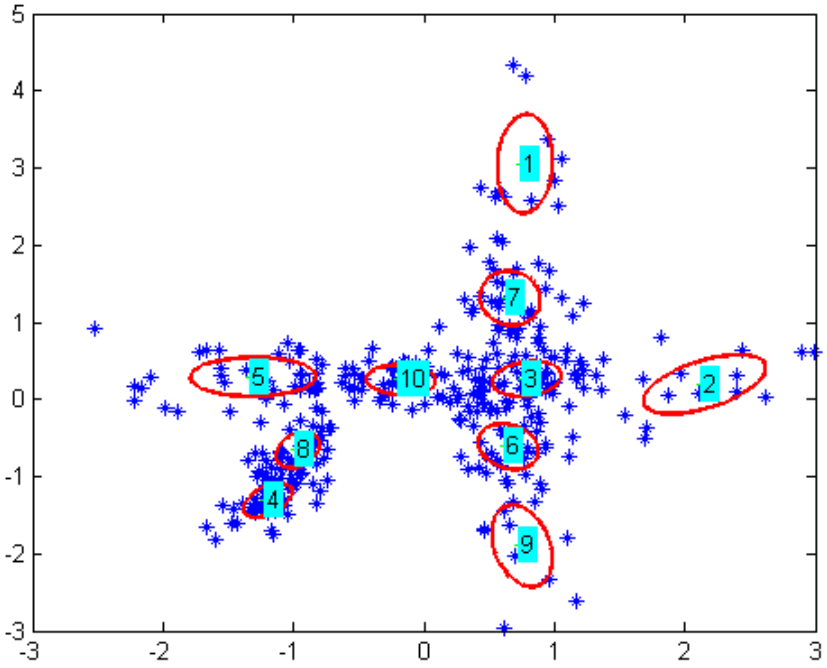
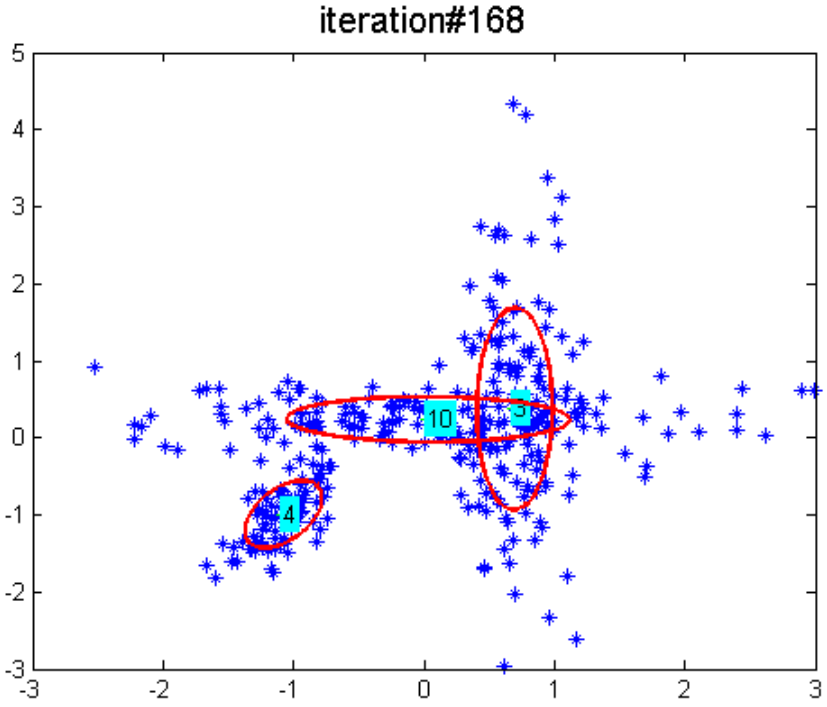


FIGURE 4.1: The synthetic 2D Gaussian mixture data



(a)



(b)

FIGURE 4.2: MAP EM result (a) shows the initialization from K-means (K=10) (b) is the MAP EM result at the converged iterations.

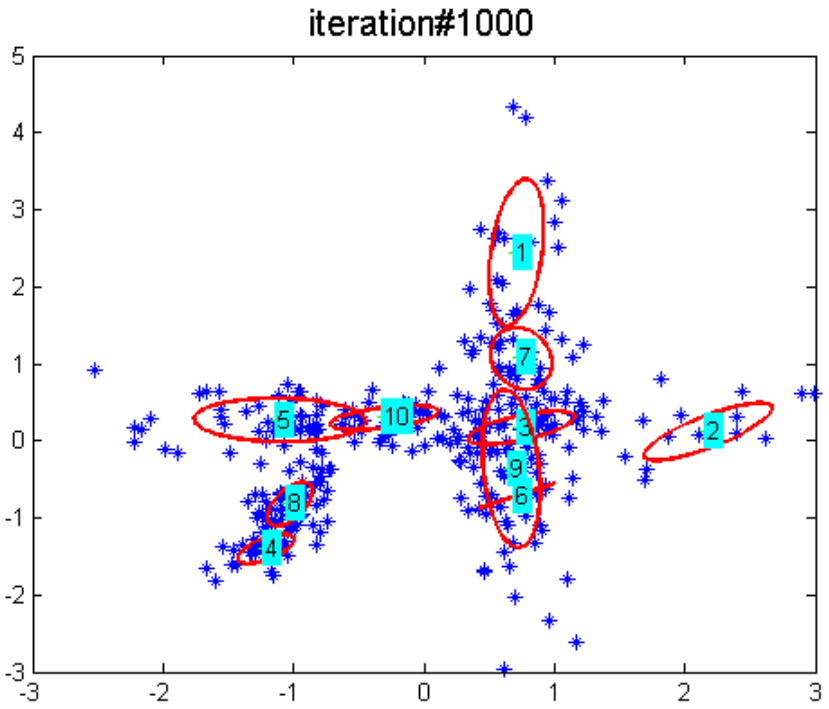


FIGURE 4.3: Regular EM result with the same initialization from K-means(K=10)

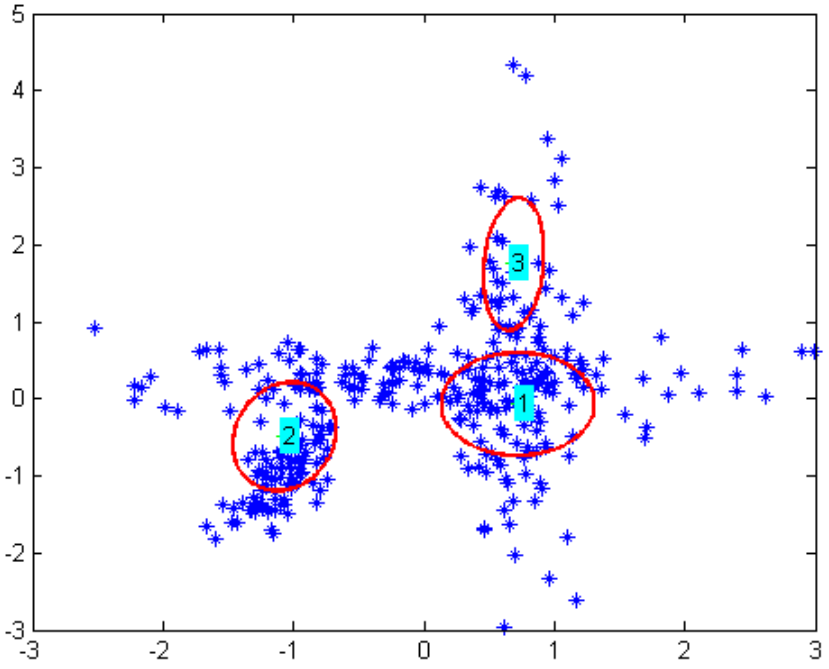


FIGURE 4.4: K-means result with K=3 classes

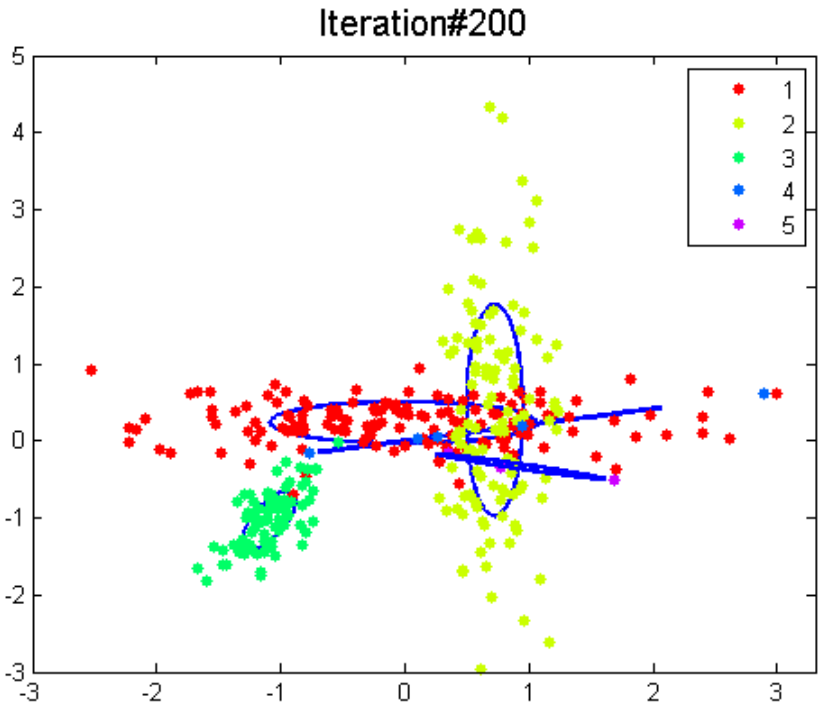


FIGURE 4.5: CRP result from initialization with only one class

Chapter 5

Objects Detection and Recognition on 3D CT Luggage Images

Computed tomography(CT) security screening is highly used in airports or other security places to detect if there is any suspicious object in traveller's luggage. It is important for preventing dangerous objects in transportation. In the CT luggage image, the number of objects is unknown, and objects are tightly packed and possible inter-wined. Also, the unavoidable metal artifacts in CT image cause the non-homogeneous of certain object. Due to these difficulties, many of state-of-the-art algorithms couldn't work very well. The image segmentation is critical, incorrectly segmentation leads to low detection rate, and decrease the travel security. Our work is to develop automatic and robust algorithms to correctly segment objects in the CT luggage, and improve the threats detection rate.

First, some state-of-the-art image segmentation methods are described. Second, we introduce our proposed MAP EM MRF segmentation algorithm. Third, our experimental results are shown. Forth, we applied machine learning techniques to classify each object into targets or non-targets.

5.1 Unsupervised Clustering and Segmentation for Objects Detection on 3D CT Luggage Images

There are a lot of segmentation algorithms available today. The commonly used image segmentation algorithms[6] are: edged-based, region-growing, graph cut and pixel-based methods. Edge-based segmentation uses the edge detecting operators which are measuring the location of the discontinuous in color, texture, etc., and then apply the post-processing to combine the edges into a border. However, the edge-based method is very sensitive to noise, and it may include "unsuitable information" in images, for example, the edge in an image is the border of the object. Region-growing segmentation is based on the homogeneity of the image. Region-growing is better in noisy images, robust and easier than the edge-based method to detect the borders of the objects' region. But, the segmentation result may have under-segmentation or over-segmentation problems. So the merging or splitting criterion is needed in post-processing. Graph cut segmentation is based on the graph $G = (V, E)$, and it is proved to achieve global optimum for binary labeling(foreground/background image segmentation). However, it may not segment well when there are multiple objects and they are not clearly different from the background. And the proper seeds are required manually. Pixel-based segmentation is to classify each pixel into one class, and the spatially connected pixels are formed into a region. The problem is that the number of classes and the parameters for each class are needed to be specified as initials.

In order to choose a proper segmentation algorithm, first, we take a look at a real CT luggage image. The original 3D CT luggage image has 512x512x293 voxels. Each voxel has its intensity value which reflects the natural properties of the object. In Figure 5.1, we took a 2D slice (512x512) at the frame 91/293 from the image. From the slice image, we can see that objects are closely packed together and the intensities in the object are not homogeneous due to the effects of the metal artifacts. Because the real CT image is

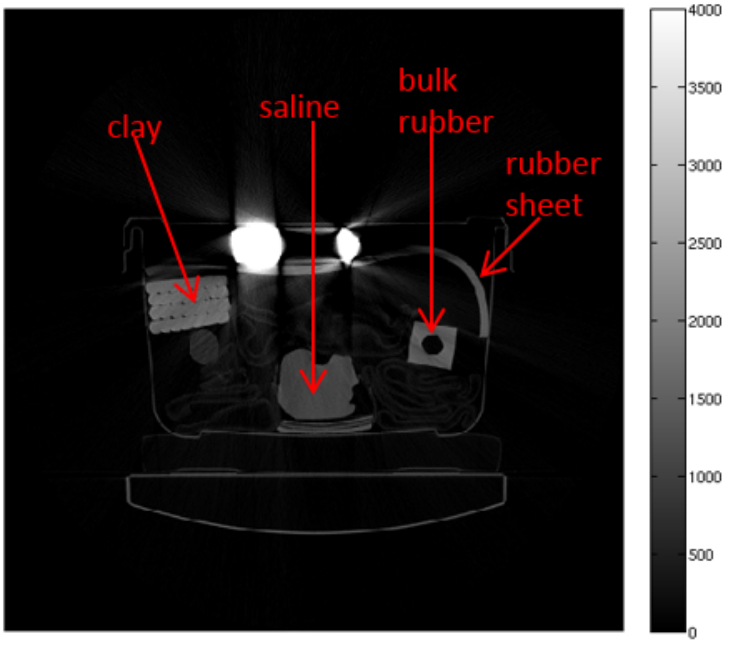
very complex as described above, the available segmentation algorithm can not be just simply applied and achieve the best result. We need to choose the proper segmentation algorithm according to the property of real CT images and also modify it with domain knowledge.

5.2 Pixel-based Segmentation with Initial Parameters from MAP EM

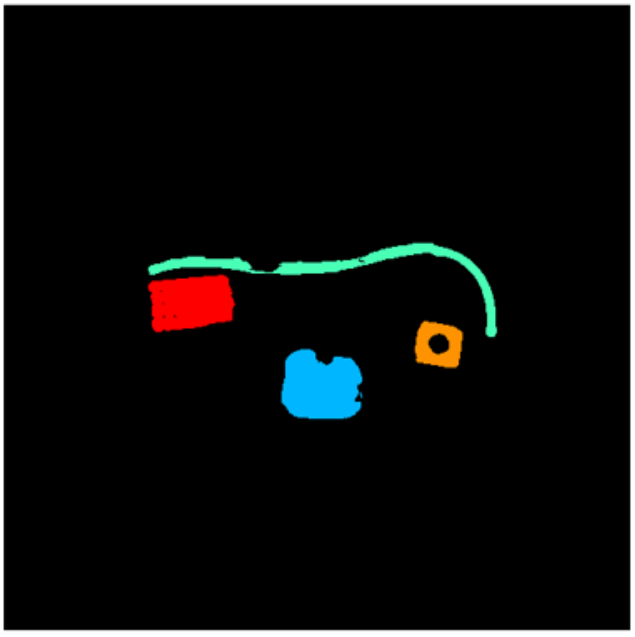
Before introducing our approach to the CT luggage image segmentation problem, we formulate the problem that we want to solve.

5.2.1 Intensity Histogram of CT Luggage Image

For example, we take a CT luggage image to analyze. In this luggage, there are 5 targets and one pseudo-target manually labeled by experts. For viewing purpose, we only plot targets in Figure 5.2. The targets are labeled as the following: 6001 saline, 6051 clay, 6012 saline, 6047 bulk rubber and 6018 rubber sheet, and the pseudo-target is labeled as 6042 saline. The intensity histogram of the whole CT luggage image is shown as the black curve in Figure 5.3 (a), and the others under the black curve are the histogram of each target which is labeled manually by experts. The gap between them is from the intensities of other non-targets objects. From the close-up view in Figure 5.3 (b), we can see the intensity histogram of each target object is close to the Gaussian distribution. Because the intensity histogram of label 6018 (rubber sheet) is significantly affected by the metal artifacts, part of the shape is different to the shape of Gaussian distribution. The overall intensity histogram in the black curve is the sum of all targets and non-targets.

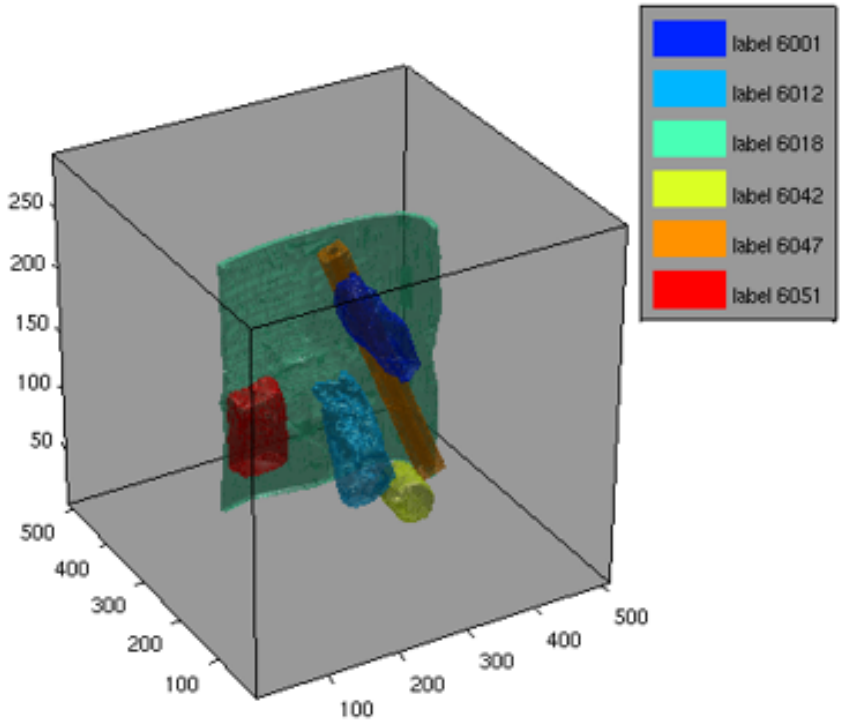


(a)

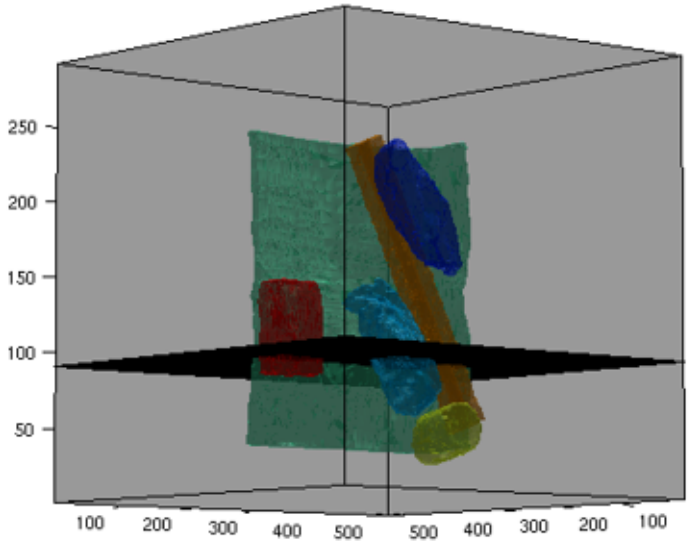


(b)

FIGURE 5.1: (a) One CT slice from 3D luggage CT image with 4 targets (saline, bulk rubber, rubber sheet and clay) (b) is the corresponding targets in (a) that manually labeled by experts.

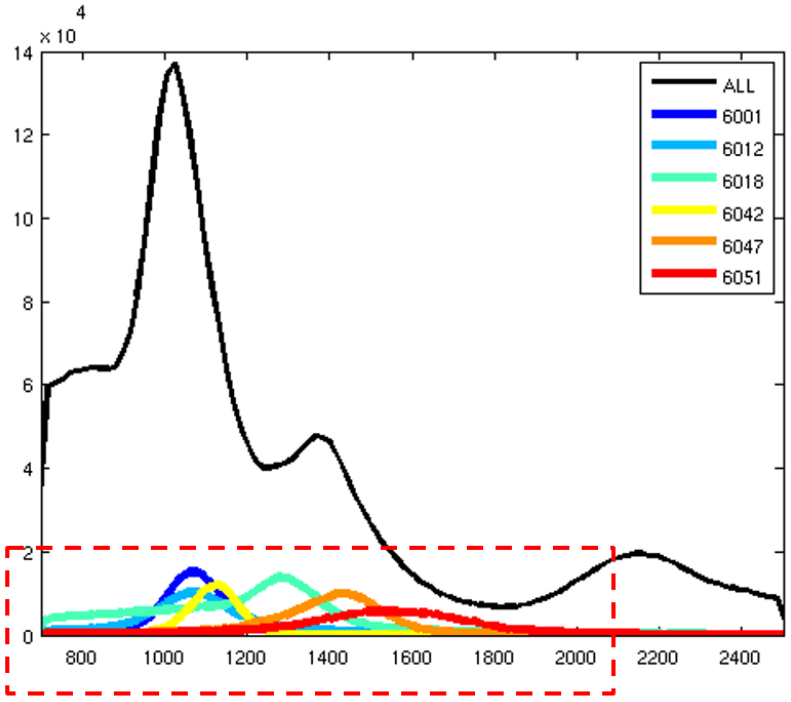


(a)

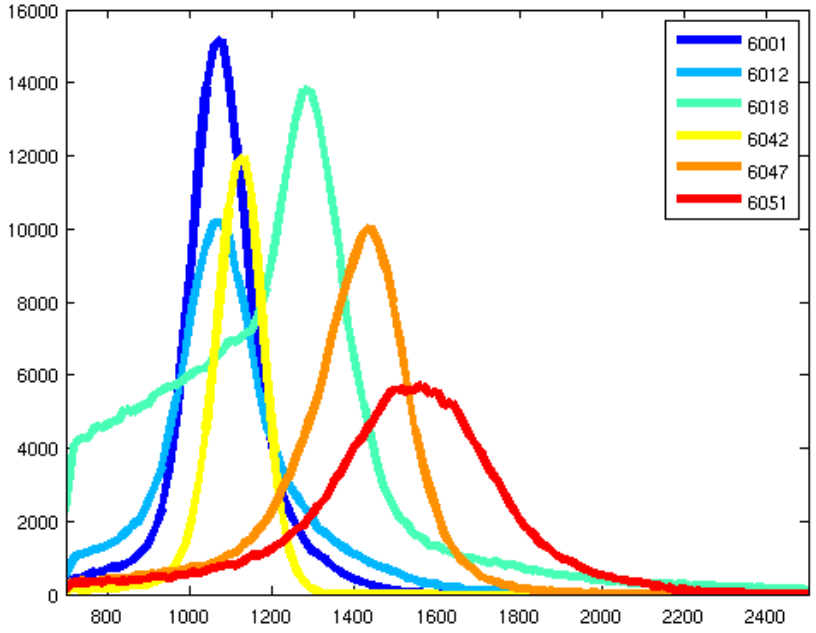


(b)

FIGURE 5.2: (a) The view of all target in 3D CT luggage image (b) is another view of (a) with the 2D black slice which is the same frame of Figure 5.1 in original CT image



(a)



(b)

FIGURE 5.3: The intensity histogram of objects in the 3D CT luggage image (a) The black curve is the histogram of the all intensity in range [700 2500]. The colored curves are the intensity histogram for each target which is corresponding to the labels in (b) is the close-up view of the targets histogram in (a).

5.2.2 Assumptions

Our assumption is that the intensity distribution of each object in the CT luggage image is a Gaussian distribution. The whole luggage can be represented as a Gaussian Mixture with different proportions. Based on this assumption, from a intensity distribution of a given CT luggage image, we use the MAP EM with Gaussian Mixture Model(GMM) to estimate the number of objects(classes) and the parameters for each object(class). Then, we use it as initials input to our pixel-based Markov Random Field(MRF) segmentation algorithm [26].

From the intensity distribution of the whole image, the MAP EM with GMM can automatically determine the number of objects and the parameters for each object. This process can solve the problem that the number of classes and the parameters for each class are needed to be pre-specified as initial conditions for pixel-based segmentation.

5.2.3 Implementations

The overall approach for the 3D CT image segmentation process can be described as follows. First, we select the intensity ranges for all the targets, and others are considered as background, and downsample. Second, we start the MAP EM algorithm with initial conditions from the Kmeans result which uses a large number of classes as input. After initialization, we set the prior parameters with our experimental analysis. Then, the MAP EM is started iteratively until the convergence condition is satisfied. At the end, after removing the extinguished classes, we obtain the number of objects and the parameters of each object, and then start the pixel-based MRF segmentation algorithm. After the objects are segmented, the mass and size of each object can be measured. If the object exceeds the max values of mass and size (e.g. mass>500 and nvoxel>300000), we consider it as a "merged" object, which need to be split by going

through the segmentation process again. The whole segmentation can be considered as two parts: coarse/1st-layer segmentation process and fine/2nd-layer segmentation process, described as the following two algorithms.

Coarse/1st-layer Segmentation Process:

Purpose: To segment object in 3D CT image

Input: 3D luggage CT image

Output: Segmented objects in image

1. Read a 3D CT image (e.g. 512x512x293).
2. Select intensity range in [700 2500], others are considered as background, and then downsample (e.g. by 16x16x4 for 512x512x293 image).
3. Apply MAP EM with GMM clustering on intensity distribution of whole image from 2:
   (1) The input is the Kmeans clustering result with K=10.
   (2) MAP EM with priors: $m0$, $S0$, $v0$, $\beta 0$.
   (3) The output is the remaining number of classes/objects with parameters(μ, σ) for each class.
4. Apply MRF segmentation process:
   (1) The input is the parameters of each class/object from 3.
   (2) The output is the set of segmented objects.
5. Apply connected component labelling to relabel the segmented objects, upsample.
6. Apply Pruning process:
   (1) The objects with small mass will be kept.
   (2) The objects with a large size and mass are treated as "merged" objects.

Fine/2nd-layer Segmentation Process:

Purpose: To split merged object from Coarse/1st segmentation process

Input: The labeled merged object

Output: Split objects

1. Read labeled merged object.
2. Apply clustering algorithm on the intensity distribution of "merged" object:
   (1) The input is the Kmeans clustering result with K=5.
   (3) The output is the remaining number of classes/objects with parameters(μ, σ) for each class.
3. Apply MRF segmentation process:
   (1) The input is the parameters of each class/object from 2.
   (2) The output is the split objects.
4. Apply connected component labelling to relabel the segmented objects.

5.2.4 Advantages of Our Approach

There are several advantages to use the two layers pixel-based MAP EM MRF segmentation approach. First, Our pixel-based MAP EM MRF segmentation approach can automatically determine the number of classes/objects, and avoid/improve the local maximum by using prior information. Second, the method can be directly applied to the initial raw CT images without any denoising pre-processing of the image. Third, The pixel-based approach can segment objects of any shape. Forth, using the two layers segmentation process, i.e. coarse/1st-layer segmentation and fine/2nd-layer segmentation, it can help to solve the under-segmentation problem. In particular, using the fine/2nd-layer segmentation can split the merged object.

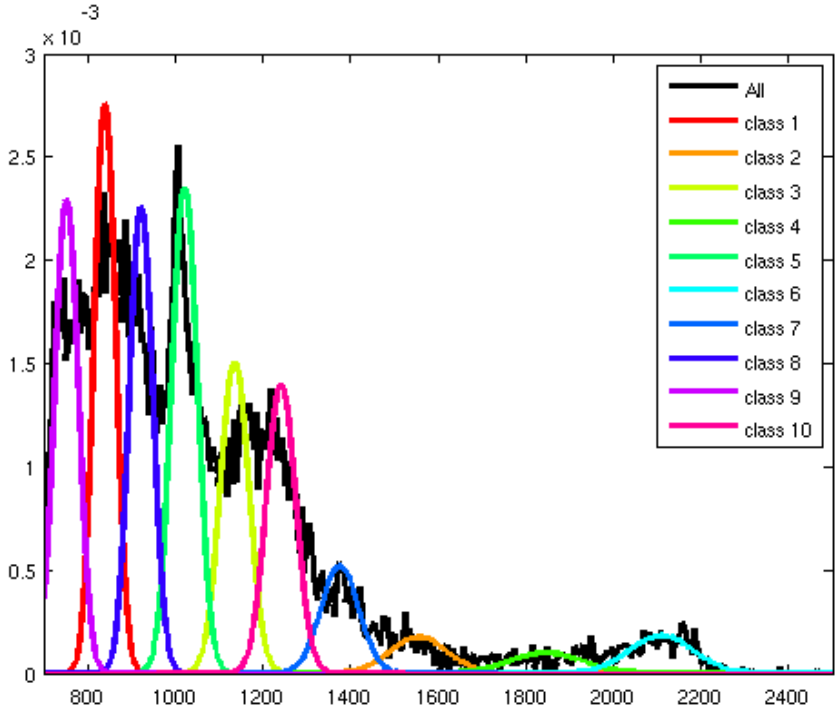
5.2.5 Segmentation Details

Now, we will show how the clustering process works on intensity distribution of an 3D CT image. First, The Kmeans (K=10) clustering result on the selected range and downsample of Figure 5.2 is shown in Figure 5.4 (a). With initial Kmeans clustering result, the result from MAP EM clustering is shown in Figure 5.4 (b). As we can see, there are five remained class(class 1, 2, 5 6 and 10), the other five classes (class 3, 4, 7, 8 and 9) are extinguished. After applying the MAP EM GMM to the intensities distribution, there are only 5 classes left from the initial 10 classes produced by Kmeans. The clustering results, i.e. the estimated parameters for each class, were used as initials input to the MRF segmentation algorithm. The MRF segmentation result is shown in Figure 5.5 (a). Among the segmented objects, we also have the target objects(labeled as 2, 4, 6, 7, 9, 10) and non-target objects(labeled as 1, 3, 5, 8, 11, 12, 13). All of detected objects are plotted in the histogram in Figure 5.5 (b). All of the targets and pseudo-targets are detected in Table 5.1 corresponding to the labels in Figure 5.5. The precision and recall for an object is calculated as the followings by using a manually labeled image(ground truth) and the detected labeled image.

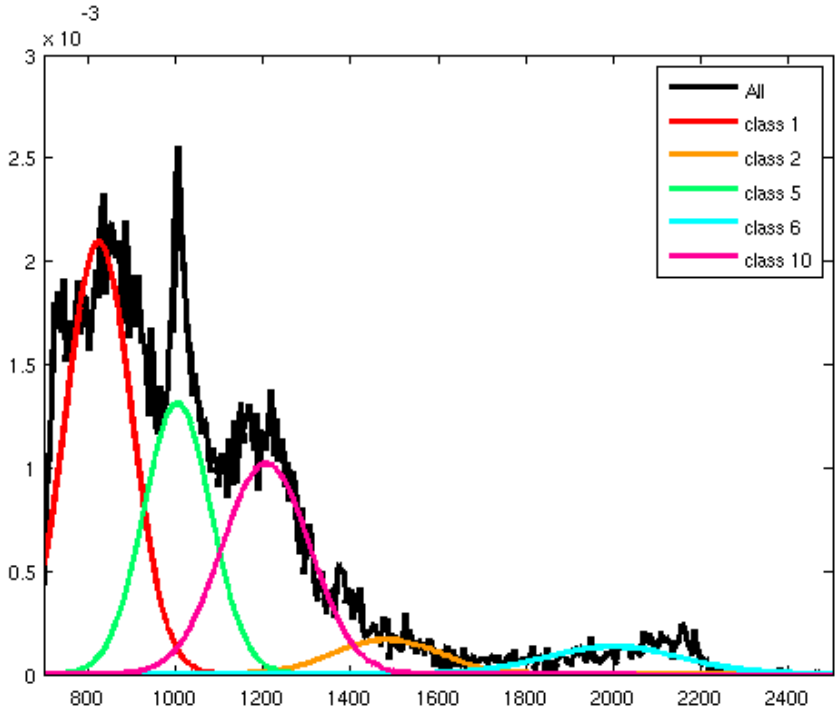
$$precision = \frac{\{ground\ truth\ label\} \cap \{segmented\ label\}}{\{segmented\ label\}} \tag{5.1}$$

$$recall = \frac{\{ground\ truth\ label\} \cap \{segmented\ label\}}{\{ground\ truth\ label\}} \tag{5.2}$$

From the equation, a low recall score is due to the fact that the detected object is oversegmented and contains only part of the ground truth object, and a low precision score is that the detected object is undersegmented and due to the fact that the detected object is merged with some fragments from other object. From the precision and recall, we can evaluate how well the object is segmented. A detected target object is considered as a target only if its precision and recall are larger than a threshold(e.g.

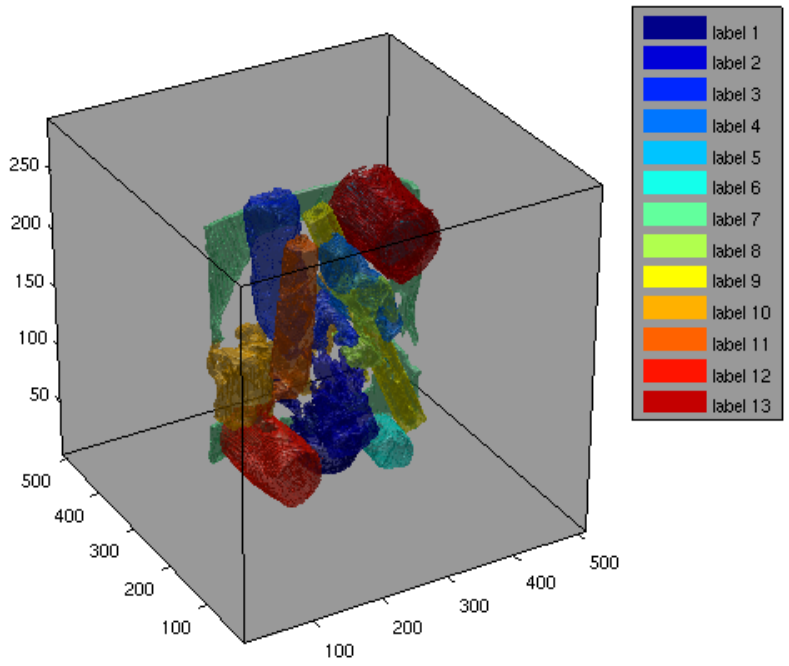


(a)

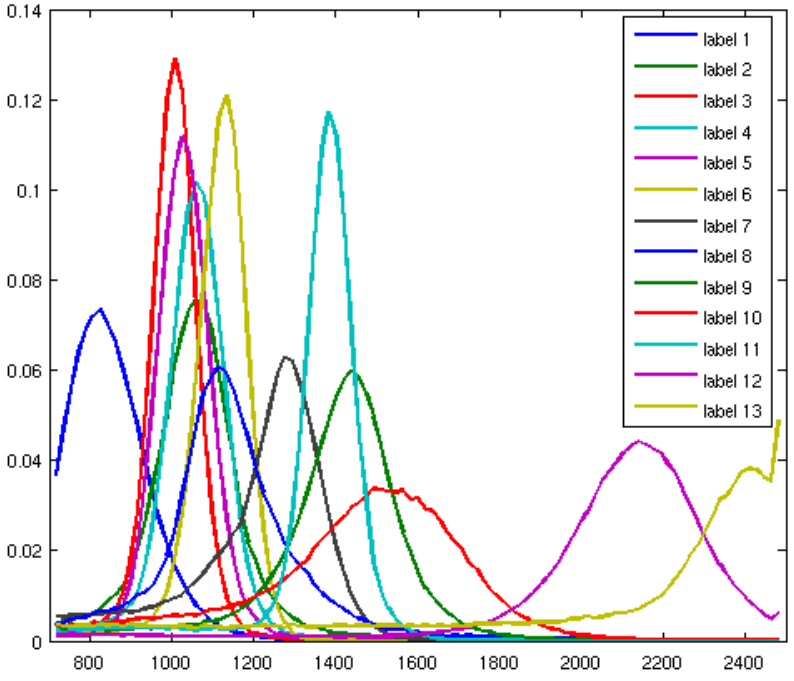


(b)

FIGURE 5.4: The histogram of the classification results from MAP EM (a) is the initialization from Kmeans by using K=10. (b) is the classification results of MAP EM



(a)



(b)

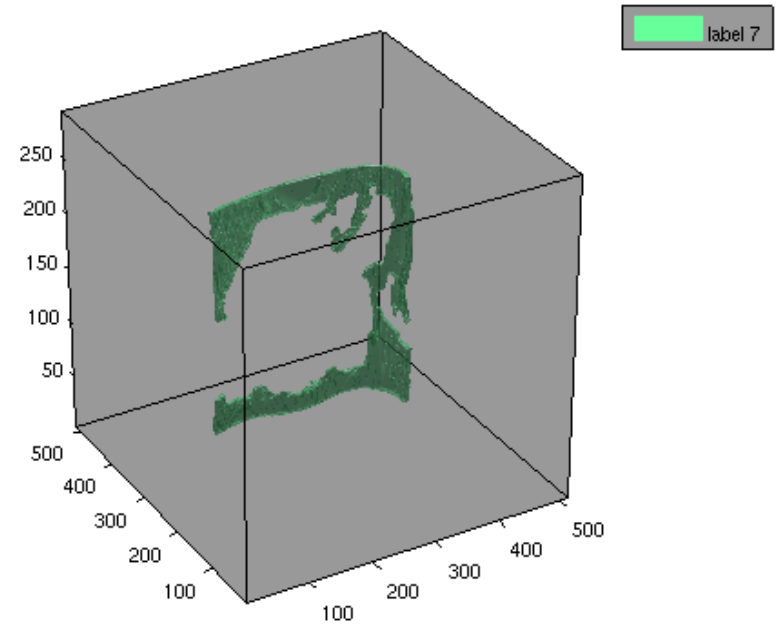
FIGURE 5.5: The segmentation results from using the MAP EM classifications as input to the MRF segmentation algorithm

TABLE 5.1: Targets Detection Summary

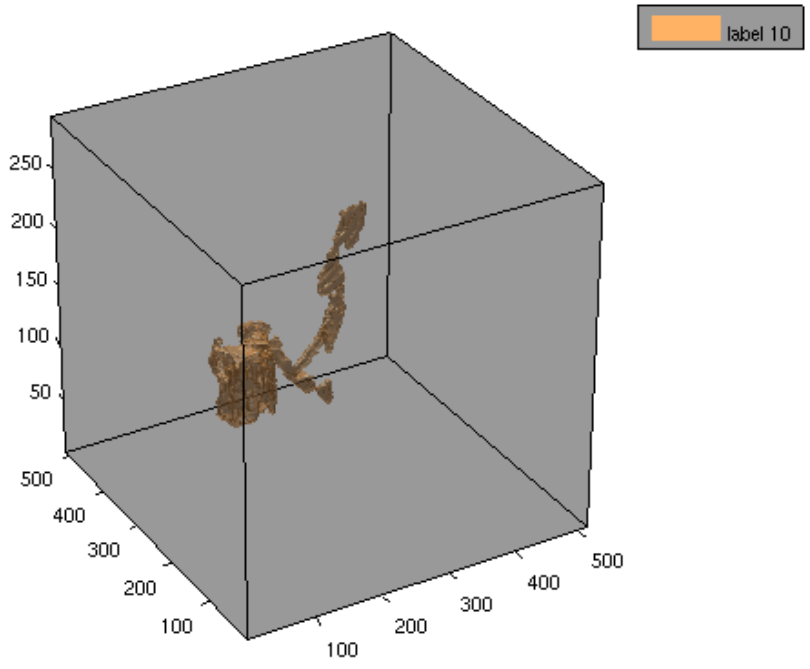
Label	Target Name	Target ID	Precision	Recall
2	saline	6012	0.91	0.58
4	saline	6001	0.90	0.57
6	saline(pseudo)	6042	0.94	0.81
7	rubber sheet	6018	0.84	0.30
9	bulk rubber	6047	0.85	0.74
10	clay	6051	0.72	0.80

0.5 except rubber sheet 0.3). From Table 5.1 and Figure 5.5, we can see the rubber sheet (label 7) has 0.30 recall. Because the metal artifacts make its intensities non-homogeneous, the rubber sheet is oversegmented, and broken into many parts, as shown in Figure 5.6 (a). The largest part only contains 30% area of labeled "rubber sheet"(ground truth) which is shown as label 6018 in Figure 5.2 (a). On the contrary, the clay (label 10) is undersegmented, as shown in Figure 5.6 (b), because some fragments from other object is incorrectly classified into the clay object. And the clay (label 10) has 0.72 precision as shown in Table 5.1.

As we can see, low precision is caused by undersegmentation. The undersegmented object could be considered as merged object. Different objects could be merged together when they have the similar characteristics and are adjacent to each other. For example, there is a merged object, shown as label 4 in Figure 5.7 (a). Inside this merged object(precision:0.49, recall:0.97), there are two objects, i.e. bulk rubber and bulk clay. To split the merged object, we make a criterion to test each object after the first segmentation process. If a object's mass and size are larger than a threshold, e.g. mass>500 and nvoxel>300000, the object is considered as a merged case, and needs to be further split. In this case, the fine/2nd-layer segmentation process is used to further split this merged object. Figure 5.7 (b) shows the result after the fine/2nd-layer segmentation. In order to analyze and visualize this merged object, the intensity histogram of the objects is plotted for the clustering and segmentation process.The



(a)



(b)

FIGURE 5.6: Illustration of low recall object (oversegmentation) and low precision object (undersegmentation)

intensity histogram of merged object Figure 5.8. The yellow curve shows the intensity histogram of label 4 in Figure 5.7(a), The green curve and the orange curve are the intensity histogram of label 4 and 5 in Figure 5.7(b). From the intensity histogram, we can see that in a merged object the intensity distributions of different objects are very close to each other, and it is very difficult to segment them using only coarse/1st-layer segmentation process. The fine/2nd-layer segmentation process is important for further splitting the merged object which could have the targets in it. And another similar example is shown in Figure 5.9(a)and(b), the histogram of the merged and segmented object is shown in Figure 5.10.

5.3 Experimental Results on 3-D CT Luggage Images

Given the ground truth and the our segmentation results, PD and PFA are the overall probability of detection and probability of false alarm by using the following equations.

$$PD = \frac{\#\ detections}{\#\ targets} \tag{5.3}$$

$$PFA = \frac{\#\ false\ alarms}{\#\ non-targets} \tag{5.4}$$

The total number of 3D CT luggage image is 188. The ground truth of the total number objects are 1850, including: 407 targets, 73 pseudo-targets, and 1370 non-targets.

5.3.1 Coarse/1st-layer Segmentation Results

From the Coarse/1st-layer segmentation, the total segmented objects are 1364: 362 targets, 38 pseudo-targets, and 961 non-targets. Based on the Eq. 5.3 and Eq. 5.4, the

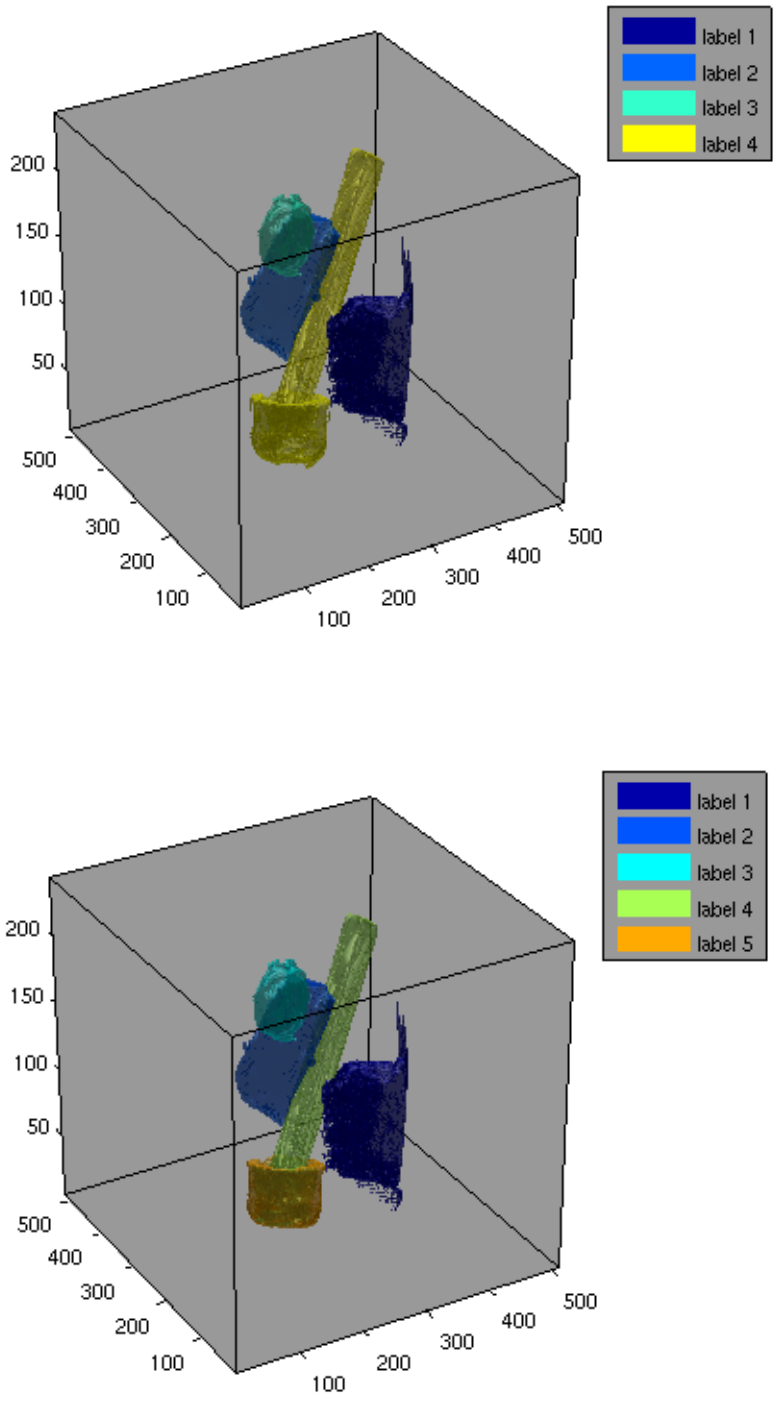


FIGURE 5.7: The figures show how the merged objects can be split during two layers segmentation. (a) shows the merged object of label 4 after the coarse/1st-layer segmentation process, (b) shows the merged object segmented into label 4 (bulk rubber) and label 5(bulk clay) after the fine/2nd-layer segmentation process

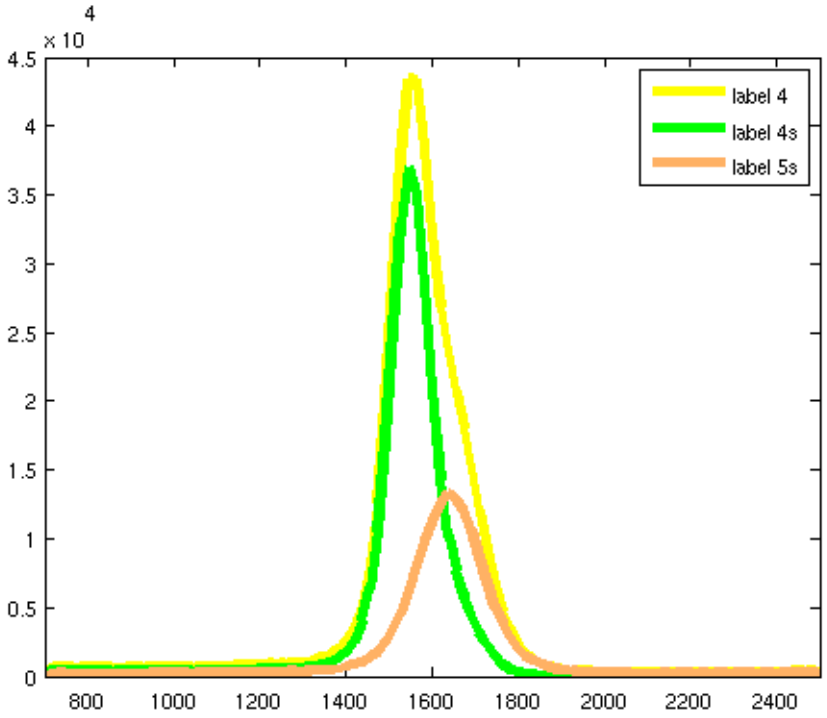
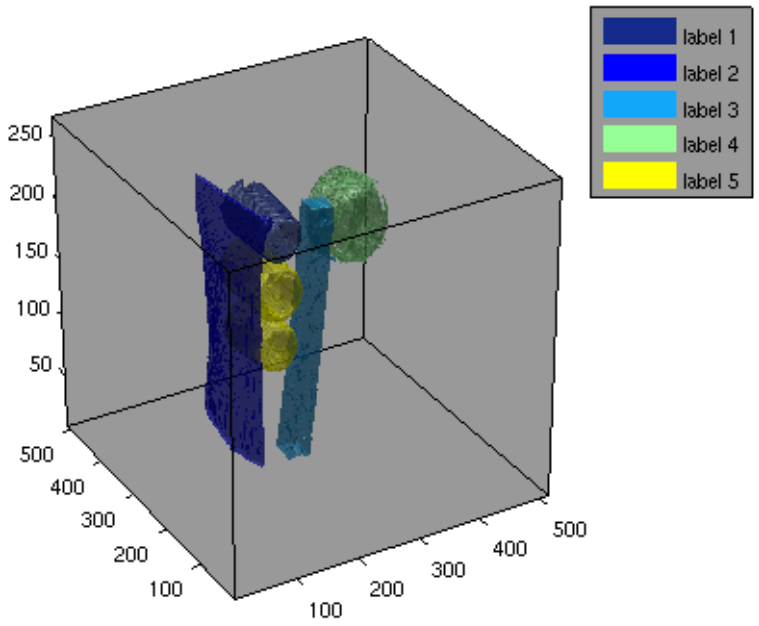


FIGURE 5.8: The intensity histogram of merged object Figure 5.7. The yellow curve shows the intensity histogram of label 4 in Figure 5.7(a), The green curve and the orange curve are the intensity histogram of label 4 and 5 in Figure 5.7(b))

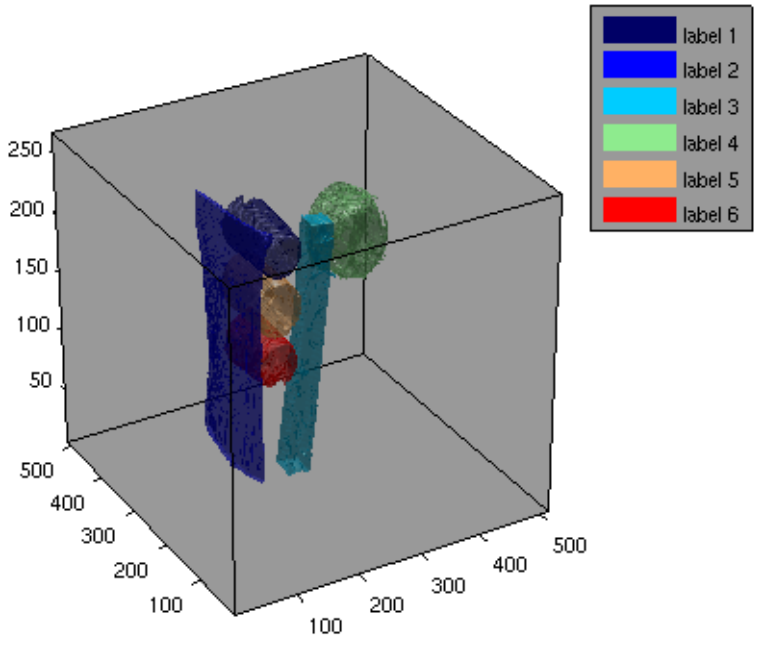
overall target PD is 89%, pseudo-target is 52%, and PFA is 70%. The details are in Table 5.2.

5.3.2 Fine/2nd-layer Segmentation Results

From the Fine/2nd-layer segmentation, the total segmented objects are 1581: 382 targets, 39 pseudo-targets, and 1160 non-targets. Based on the Eq. 5.3 and Eq. 5.4, the overall target PD is 94%, pseudo-target is 53%, and PFA is 85%. The details are in Table 5.3. Compared with the Coarse/1st-layer segmentation, the Fine/2nd-layer segmentation increases the target detection rate (PD) by 5%, however the false alarm rate(PFA) increases higher by 15%.



(a)



(b)

FIGURE 5.9: Another example shows how the merged saline can be split during two layers segmentation. (a) shows the merged object (label 5) after the coarse/1st-layer segmentation process, (b) shows the merged object segmented into two objects (label 5 and label 6) after the fine/2nd-layer segmentation process

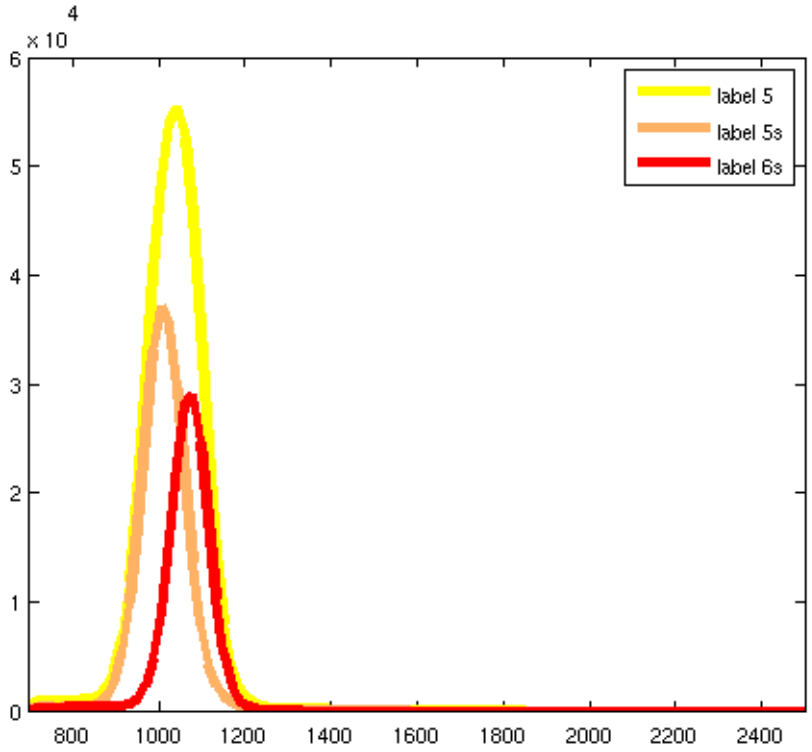


FIGURE 5.10: The intensity histogram of merged object Figure 5.9. The yellow curve shows the intensity histogram of label 5 in Figure 5.9(a), The orange curve and the red curve are the intensity histogram of label 5 and 6 in Figure 5.9(b))

5.4 Supervised Classification for Target Recognition in 3D CT Luggage Images

Using the two-layers segmentation, we successfully increases the target detection rate (PD) by 5%, however the false alarm rate(PFA) increases higher by 15%. In this section, we will describe our work in using machine learning techniques to reduce false alarm rates, without reducing the targets detection rate too much. A merged object is not only the targets merged with target or non-target, but also having the non-targets merged together. Because the merged object is caused by these objects have the similar characteristics and adjacent to each other. Hence, splitting the merged object can detect more target objects, but also increase the number of non-target object. Due to

TABLE 5.2: Coarse/1st-layer Segmentation Results

Type	**Subtype**	**Difficulty**	**# Targets**	**# Detected**	**PD%**
Target	All	All	407	362	88.9
Target	Clay	All	111	104	93.7
Target	Rubber	All	158	142	89.9
Target	Saline	All	138	116	84.1
Target	Bulk	All	270	234	86.7
Target	Sheet	All	137	128	93.4
Target	All	Low	77	73	94.8
Target	Clay	Low	29	29	100
Target	Rubber	Low	22	22	100
Target	Saline	Low	26	22	84.6
Target	Bulk	Low	56	52	92.9
Target	Sheet	Low	21	21	100
Target	All	High	317	277	87.4
Target	Clay	High	82	75	91.5
Target	Rubber	High	125	110	88
Target	Saline	High	110	92	83.6
Target	Bulk	High	201	170	84.6
Target	Sheet	High	116	107	92.2
Pseudo-Target	All	High	73	38	52.1
Pseudo-Target	Clay	High	10	10	100
Pseudo-Target	Rubber	High	10	8	80
Pseudo-Target	Saline	High	19	11	57.9
Pseudo-Target	Powder	High	34	9	26.5
Pseudo-Target	Bulk	High	63	30	47.6
Pseudo-Target	Sheet	High	10	8	80

the effect of metal artifacts, some targets intensity is lower than their normal values. In order to detect targets as much as possible, we select the intensity range in [700, 2500], and the others are considered as background. Meanwhile, it is unavoidable to include more non-targets in our segmentation results. Increasing the detection rate, i.e. increasing the security, is very important, and the non-targets/false alarm can be also be filtered by using supervised classification method.

TABLE 5.3: Fine/2nd-layer Segmentation Results

Type	Subtype	Difficulty	# Targets	# Detected	PD%
Target	All	All	407	382	93.9
Target	Clay	All	111	106	95.5
Target	Rubber	All	158	148	93.7
Target	Saline	All	138	128	92.8
Target	Bulk	All	270	251	93
Target	Sheet	All	137	131	95.6
Target	All	Low	77	73	94.8
Target	Clay	Low	29	29	100
Target	Rubber	Low	22	22	100
Target	Saline	Low	26	22	84.6
Target	Bulk	Low	56	52	92.9
Target	Sheet	Low	21	21	100
Target	All	High	317	297	93.7
Target	Clay	High	82	77	93.9
Target	Rubber	High	125	116	92.8
Target	Saline	High	110	104	94.5
Target	Bulk	High	201	187	93
Target	Sheet	High	116	110	94.8
Pseudo-Target	All	High	73	39	53.4
Pseudo-Target	Clay	High	10	10	100
Pseudo-Target	Rubber	High	10	7	70
Pseudo-Target	Saline	High	19	13	68.4
Pseudo-Target	Powder	High	34	9	26.5
Pseudo-Target	Bulk	High	63	32	50.8
Pseudo-Target	Sheet	High	10	7	70

5.4.1 Data Features

Feature selection [51, 52] is an important step in data mining, and it could affect the performance of classifiers. The data set usually includes many non-informative/noisy or redundant features which are not useful for the training process. Reducing those features could improve classification accuracy. Feature selection could reduce the overfitting, and also decrease training time by removing the redundancy.

The feature selection has two types. First, feature variables can be dependent or independent to each other. Second, feature selection can be dependent or independent

to the classifier/classification model. The combination from each type can form different feature selection techniques. For example, feature selection can be independent of the classifier and independent between feature variables. There are several ways to perform feature selection, e.g. mutual information, information gain, etc. Or feature selection can be correlated with the classification models. For some classifier, to get better accuracy, feature vectors need to be normalized, e.g. as in SVM algorithm, or feature variables can be transformed into non-linear form in classification model. The purpose of feature selection is to find the more predictable variables. Before we apply the classification algorithm to our targets detection in 3D CT luggage image, we need to extract and evaluate the data features.

In a CT image, the pixel/voxel intensity values represent the physical and chemical properties of the objects. And based on the segmentation results, we can extract various features from the segmented regions in CT image. These features can be used to classify the regions into targets or non-targets. From the segmented regions, based on the intensity values, we can calculate the density, mass, number of voxels, average of gradient, mean, mode, variance, and histogram of each region. The intensity histogram of the segmented region is constructed by 100 bins in this work, and each bin can be represented as one feature. So the total number of features vector is a 107 dimension variables.

5.4.2 Model Selection

Cross Validation [53] is a model/classifier selection and evaluation method. It is a process for tuning/finding the "optimal" parameters of a selected model and evaluating the performance of the model: how well it can be generalized to new/unseen data. Usually, the classifier can make accurate predication on the data which appears in its training data set, but can make wrong decisions on new/unseen data. One way to

overcome this overfitting problem is to use a subset of the entire data as the training data, and the rest can be used to test the performance of the trained model. K-fold cross validation randomly divides the data set into K mutually exclusive subsets(folds) of approximately equal size. Each time, one of the K fold is used as test set, the others ((K-1)folds) are used as the training set. The cross validation is repeated K times, and all the data is used for both training and testing. At the end, the average error rate will be estimated across the K folds. In reality, the choice of the number of folds is dependent on the size of available data set. In our experiments, we tested with 3, 5, and 10 fold cross validation.
The procedure is:

1. Load the data, and split the data into training and test sets.
2. Select a model with parameters.
3. Train and evaluate the model in K-fold cross validation.
4. Tune the parameters and repeat step 3.
5. Select the best model and assess this final model in test/all data set.

The performance of the classifier from cross validation process is evaluated by using the Receiver Operating Characteristic (ROC) curve. The ROC is a plot of true positive rate(TPR, sensitivity, recall) against the false positive rate(FPR, type I error, 1-specificity). The best prediction is a point in the upper left corner or close to 100% sensitivity and 100% specificity. It shows the tradeoff between the sensitivity and specificity, i.e. an increase in sensitivity is accompanied by a decrease in specificity. We will use the ROC curve in K-fold cross validation to evaluate the performance of classification models and select the best model with its parameters for our target detection in 3D CT luggage images.

5.4.3 Experimental Results

Our experiments are performed on 188 3D CT luggage images. The ground truth, which are manually labeled by experts, includes 407 targets, 73 pseudo-targets, and 1370 non-targets. In Chapter 5, our best segmentation in 188 3D CT luggage images resulted in a classification performance with 94% PD and 85% PFA. The total segmented objects are 1581: 382 targets(threats), 39 pseudo-targets, and 1160 non-targets (non-threats). In this chapter, our objective is to detect the targets from the non-targets by using supervised classification method[54]. The classification results are represented as PD and PFA which are calculated based on the Eq. 5.3 and Eq. 5.4. Our two-layers segmentation results contain 1542 detected objects, including 382 targets and 1160 non-targets. For segmented object recognition, we applied machine learning techniques to classify each object into targets or non-targets, and selected the best AUC score as our best classification method. Because of the security requirement by a DHS project, we need to maintain the PD around 90%, and reduce the PFA as much as possible. For better comparison, along the ROC curve, we average all the PFA of each fold at the PD around 90%. The details are in Table 5.4.

TABLE 5.4: Classification Results in 10-fold Cross Validation

Two-class Classification	**10-fold Cross Validation**			
	Num Targets	**PD%**	**Num FAs**	**PFA%**
RandomForest	367	90%	235	17%
ExtraTrees	362	89%	248	18%
AdaBoost	365	90%	184	13%
GradientBoosting	364	89%	206	15%
NearestNeighbors	363	89%	255	19%
SVM	364	89%	220	16%
LogisticRegression	365	90%	300	22%
LogisticRegression(Polynomial)	364	89%	232	17%
BayesianAdditiveRegressionTrees	362	89%	267	19%
GaussNaiveBayes(log)	363	89%	332	24%

After model selection, the tuned model is trained on half dataset, and tested on all dataset. The classification results are shown in Table 5.5. As we can see, the best classifier are boosting based methods, such as AdaBoost and GradientBoosting. We have achieved good results with 90% PD and 6% PFA.

TABLE 5.5: Classification Results in Half Training and All Testing

Techniques	**Half Training and All Testing**			
	Num Targets	**PD%**	**Num FAs**	**PFA%**
RandomForest	367	90%	130	9%
ExtraTrees	367	90%	170	12%
AdaBoost	367	90%	80	6%
GradientBoosting	367	90%	83	6%
NearestNeighbors	367	90%	120	9%
SVM	367	90%	192	14%
LogisticRegression	367	90%	334	24%
LogisticRegression(Polynomial)	367	90%	221	16%
GaussNaiveBayes(log)	367	90%	405	30%

Chapter 6

Scatter Correction by Non-Local Techniques

In conventional X-ray imaging (with X-ray panels), scattered radiation can produce noise and a number of image artifacts, such as cupping, shadows, and decreased soft-tissue contrast [56, 57]. In practice, hardware solutions such as anti-scatter grids are often used to reduce scatter. However, the remaining scatter can still be significant and additional software-based corrections are often desirable. Furthermore, good software solutions can potentially reduce the amount of needed anti-scatter hardware, thereby reducing cost. Indeed, software correction algorithms have begun to appear in some commercial X-ray imaging products.

In today's more advance software techniques, scatter correction is done in two steps [56, 57]. First, a scatter model is estimated from training data. Then, the model is used in a reconstruction algorithm to produce a corrected image. Although these techniques have achieved some success, they still have some limitations that make it difficult to fully meet clinical imaging quality demands. For example, scatter models estimated from training data may not always fit particular patients well and this can lead to insufficient scatter correction. Similarly, although these algorithms often employ edge-preserving priors, such as the TV (total variation) or MRF (Markov random fields) (e.g., see [56, 61]), to maintain good local contrast, such priors generally cannot preserve

textural details, which are often important to diagnosis. Finally, these algorithms are usually complex and iterative and current implementations tend to be too slow.

The purpose of this work is to develop better software scatter correction algorithms. Towards that end, we take a broader view and observe that, mathematically, the problem of scatter correction has a lot of similarities to a more traditional but still very actively researched image processing problem, i.e., image restoration. Specifically, scatter correction amounts to estimating the original image when it is corrupted by Poisson noise and is also possibly distorted by a linear or a non-linear transformation, while image restoration amounts to estimating the original image when it is corrupted by Gaussian noise and is also possibly distorted by (usually) a linear transformation. Indeed, the aforementioned TV and MRF based techniques were initially developed for image restoration. Recently, a number of non-local techniques such as [58, 60] and their extensions have significantly improved image restoration results. Specifically, they provide better results by preserving textural details better, which is difficult by traditional or even TV and MRF techniques which are local in nature. Furthermore, recent advances in blind deconvolution (de-blurring) e.g., [62, 63] can potentially be exploited to produce better scatter model estimation. In this work, we adapt some of the non-local techniques for image de-noising and image restoration to scatter reduction and demonstrate their efficacy. In addition to X-ray imaging, our techniques could also be extended relatively straightforwardly to scatter correction in CT (computer tomography).

6.1 Method

Let $x = \{x_i\}$ and $s = \{s_i\}$, respectively, be the "true" intensity image and scatter, where x_i and s_i are, respectively, the true image pixel and scatter at location i. In X-ray

imaging, x and s are not directly observable; instead, the observed image $y = \{y_i\}$ is a Poisson random field with $x + s$ as mean. Specifically, each observed pixel y_i is

$$y_i \sim \text{Poisson}(x_i + s_i) \tag{6.1}$$

where $\sim$ denotes that y_i is a random sample from a Poisson distribution with mean $x_i + s_i$.

A simple and commonly used scatter model is (for all i)

$$s_i = s_0 \tag{6.2}$$

where s_0 is a constant average scatter level. A more complex model [57] that allows one to account for the effect of object thickness is a convolution model with

$$s_i = (x * h)_i + s_0 \tag{6.3}$$

where h is a kernel, $*$ is convolution, and $s_0 \geq 0$ is a constant. This model can also be extended to a nonlinear form [57], with

$$s_i = (u * h)_i + s_0 \tag{6.4}$$

where $u = \{u_i\}$ is related to x through

$$u_i = x_i(\log I_0 - \log x_i) \tag{6.5}$$

where I_0 is a constant max intensity.

Given a scatter model and the Poisson imaging model of (6.1), the problem of (software) scatter correction becomes: given y, estimate x. A powerful approach to solving this problem is the Bayesian MAP (maximum a posteriori) with

$$\widehat{x} = \arg\max_x \left\{ \log p(y|x) + \log p(x) \right\} \tag{6.6}$$

where $p(y|x)$ is the likelihood function and $p(x)$ is the prior. In general, $p(y|x)$ is determined by the Poisson imaging and scatter model, $p(x)$ is selected based on prior knowledge or, more often, desired properties we want the corrected images to have, and the maximization can be performed by using a relatively standard optimization algorithm. We will discuss each of these in turn, starting with $p(y|x)$.

Assume the pixels in the observed image y are conditionally independent given the true image x and scatter s; then

$$p(y|x) = \prod_i p(y_i|x, s) \tag{6.7}$$

where the product is over all pixel locations in y. In logarithm, we have

$$\log p(y|x) = \sum_i \log p(y_i|x, s) \tag{6.8}$$

From the imaging model of (6.1), $p(y_i|x, s)$ is a Poisson distribution with mean $x_i + s_i = \lambda_i$, i.e.,

$$\log p(y_i|x, s) = \log p(y_i|\lambda_i) = \log \left[\frac{\lambda_i^{y_i} e^{-\lambda_i}}{y_i!} \right] = -\log(y_i!) + y_i \log \lambda_i - \lambda_i. \tag{6.9}$$

If we view $\log p(y_i|\lambda_i)$ as a function of λ_i and approximate it by a second-order Taylor series expansion around y_i, we will have

$$\log p(y_i|x,s) = \log p(y_i|\lambda_i) \simeq c(y_i) - \frac{(y_i-\lambda_i)^2}{y_i} \tag{6.10}$$

where $c(y_i)$ is a function of only y_i (therefore a constant because y is observed) and we used the fact that

$$\frac{\partial}{\partial \lambda_i} \log p(y_i|\lambda_i) = 0 \tag{6.11}$$

for $\lambda_i = y_i$. The approximation in eqn (6.10) converts the log-likelihood function $\log p(y|x)$ into a quadratic function of λ_is and, consequently, a quadratic function of x and s and this can potentially make the MAP optimization easier. Similar approximations have been used successfully in iterative CT (computer tomography) reconstruction [59].

Now we can consider the prior $p(x)$. A popular choice in many image restoration and reconstruction applications is the total variation (TV) prior. It can be written as

$$\log p(x) = C_1 - \gamma \sum_i \sqrt{x_{i,h}^2 + x_{i,v}^2} \tag{6.12}$$

where C_1 and $\gamma > 0$ are constants and $x_{i,h}$ and $x_{i,v}$ are, respectively, the horizontal and vertical pixel difference at location i (approximations to horizontal and vertical partial derivatives). This prior tends to preserve strong edges but can smooth over texture (which often have weak edges); it also favors images with constant patches ("staircases" images). Recently, several priors were proposed to overcome this problem. One is the non-local means (NLM) prior [58], with

$$\log p(x) = C_2 - \gamma \sum_{i \neq j} w_{i,j}(x_i - x_j)^2 \tag{6.13}$$

where C_2 and $\gamma > 0$ are constants and the sum is over all pixel pairs and $w_{i,j}$ is a weight that comes from the difference between the patches that surround x_i and x_j, defined as

$$w_{i,j} = \frac{1}{\sum_j e^{-\beta \sum_k (x_{i+k} - x_{j+k})^2}} e^{-\beta \sum_k (x_{i+k} - x_{j+k})^2} \tag{6.14}$$

where $\beta > 0$ is a scale parameter and the sum is over all pixels in the patches. Intuitively, the NLM prior tends to average pixels that have similar surroundings or texture and does not average pixels that have different texture; hence, it is potentially more texture-preserving. Another prior similar to the NLM is the non-local TV (NLTV) (e.g., see [60]), defined as

$$\log p(x) = C_3 - \gamma \sum_i \sqrt{\sum_j w_{i,j} (x_i - x_j)^2} \tag{6.15}$$

where C_3 and $\gamma > 0$ are constants and the weight $w_{i,j}$ is the same as that in the NLM prior. This prior has been reported to have slightly better results than the NLM for image restoration [60] but the NLM, with its quadratic form, is easier to optimize and is preferred here.

There are three practical issues related to computing the weights for the NLM prior. The first is that to compute the weights, we need the true image x, which is unknown. One way to solve this problem is to replace the true image by an estimate, obtained separately with a different prior (e.g., with a TV prior or a non-weighted quadratic prior) and compute the weights from the estimate [60]. The second practical issue is that we need to choose the size of the patch that surrounds a pixel. Furthermore, since computing weights for all pixel pairs in an image is computationally expensive, we only compute the weights for two pixels if they are within a "search window". In this work, we followed [60] to choose the size of the patch and the size of the search window to be 5×5 and 31×31, respectively. The third practical issue is the selection of parameter β,

see eqn (6.14). In the original NLM work [58], β was set to be proportional to $1/\sigma^2$, where σ^2 is the variance of the observation noise. The idea is that higher noise will make the calculated patch-difference less reliable hence, the weights should be more "spread out." In our current work, through the quadratic approximation of eqn (6.10), the noise variance is y_i, not a constant. Hence, in our NLM based algorithm, β is location dependent and is set to be proportional to $1/y_i$.

Finally, for the maximization in our MAP approach, we used a relatively standard gradient descent algorithm.

6.2 Results

We tested our NLM prior based scatter correction algorithm on simulated and real images and some typical results are shown in Figs. 6.1 to 6.6 (to view some of the image details better, we suggest to use 200% to 400% when viewing the pdf file). For performance evaluation, we used the PSNR (peak signal-to-noise ratio) when ground truth is available; the PSNR essentially measures estimation error, the larger the PSNR, the smaller the error. When ground truth is not available, we used visual inspection and some ad hoc measures, such as the contrast before and after correction [64, 65]. Generally, the ad hoc measures tend to agree with the results of visual inspection.

In Fig. 6.1, our NLM correction algorithm was used to correct a scattered image simulated with the simplest scatter model, i.e., the constant scatter level model of eqn (6.2). For purpose of comparison, we also provided the un-scattered image (ground truth) and result obtained by using a TV prior based algorithm. In addition to images, profiles along a typical horizontal line of the images were also provided to allow another view of the results. As can be seen from Fig. 6.1, both the TV and NLM prior based algorithms generated good results but the NLM's result is better.

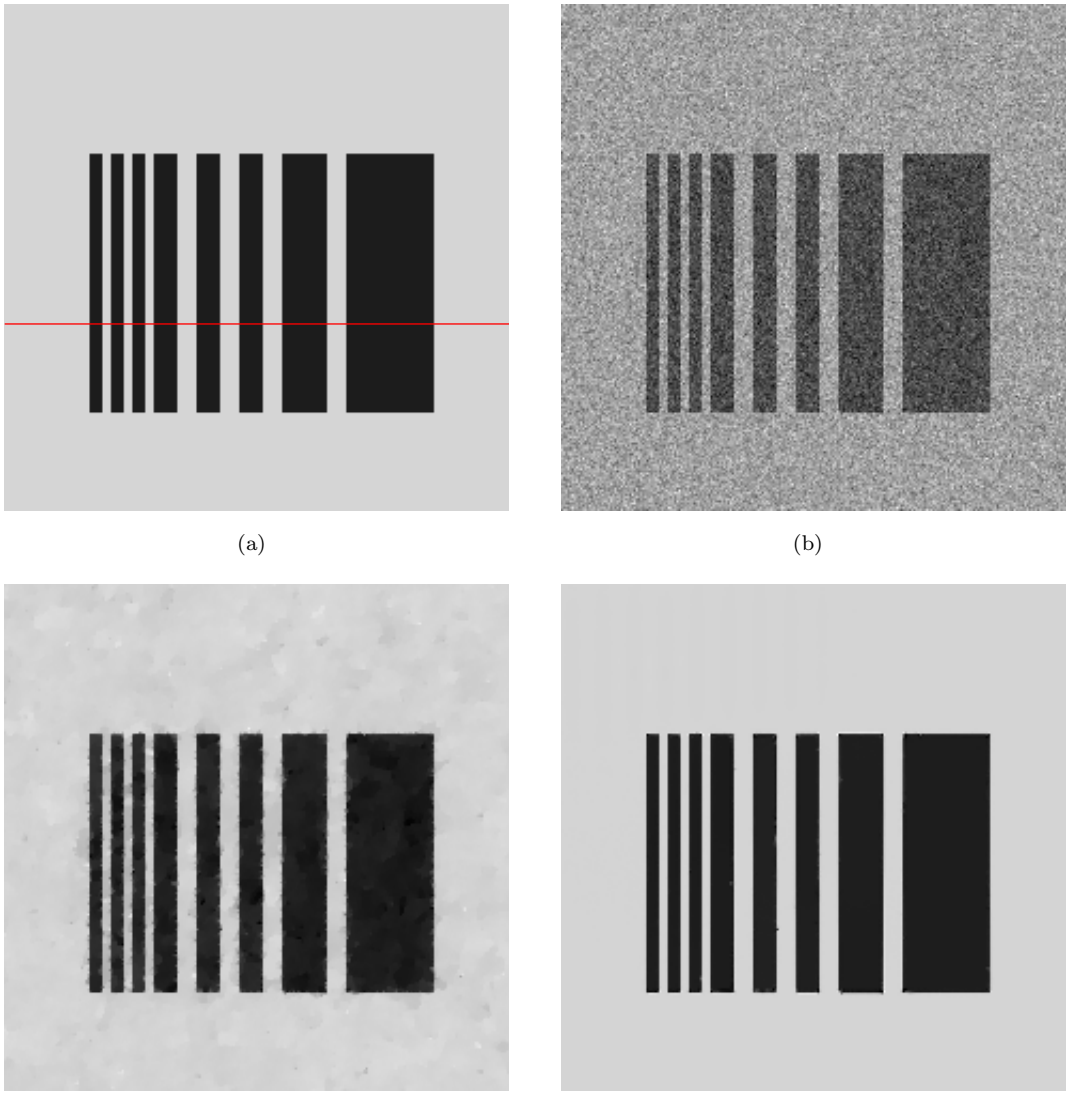
In Fig. 6.3, the experiment in Fig. 6.1 was repeated using an X-ray image of a phantom. Here the scattered image is obtained by placing a plexiglass sheet in front of the phantom while the "clinical ground truth" image was obtained by adding an anti-scatter grid in front of the X-ray sensor. Since the clinical ground truth and the corrected images have different dynamic ranges, PSNR calculation is not feasible. As can be seen in Fig. 6.3, again both the TV and NLM algorithms corrected a lot of the scatter but the result from the NLM has finer detail and does not have blocky artifacts.

In Fig. 6.4, our NLM correction algorithm was used to correct a scattered image simulated with the nonlinear convolution scatter model of eqns (6.4) and (6.5), with h a Gaussian kernel, $s_0 = 0$, and I_0 a constant corresponding to the maximum intensity level for the X-ray imaging process. Furthermore, the two-dimensional Gaussian kernel 13×13 has a variance of 4 and is scaled by a factor of 0.75. As can be seen in Fig. 6.4, the scattered image not only contains Poisson noise (speckles) but also contains blur. Both the TV and NLM algorithms generated good correction results but the latter's result is better.

In Fig. 6.6, the experiment in Fig. 6.4 was repeated using an X-ray image of an abdomen phantom. Here the scatter was "generated naturally" by the internal structures of the phantom itself. Since the effect of the scatter and scatter correction is difficult to see due to the large size of the images, we provide zoomed-in views of some of the corresponding regions. Again, both the TV and NLM algorithms were able to make the input (scattered) image sharper and improve contrast but the NLM algorithm does not produce much blocky artifacts.

6.3 Conclusion and Future Work

In this work, we adapted the non-local mean techniques originally proposed for image restoration to the problem of scatter correction. Our algorithm is based on a Bayesian MAP approach, where the likelihood function comes from scatter models and the prior comes from a non-local means prior. Experimental results indicate that our algorithm is promising for scatter correction: it reduces the effect of scatter, such as noise and blur, while preserving textural details. For future work, we plan to incorporate scatter model estimation into our algorithm and also accelerate the algorithm by multi-resolution processing.



(a) (b)

(c) (d)

FIGURE 6.1: Scatter correction on a simulated image. Scatter model: the constant scatter level. (a)ground truth; (b)scattered image; (c)result from TV, PSNR=25.57dB; (d)result from our NLM algorithm, PSNR=35.53dB.

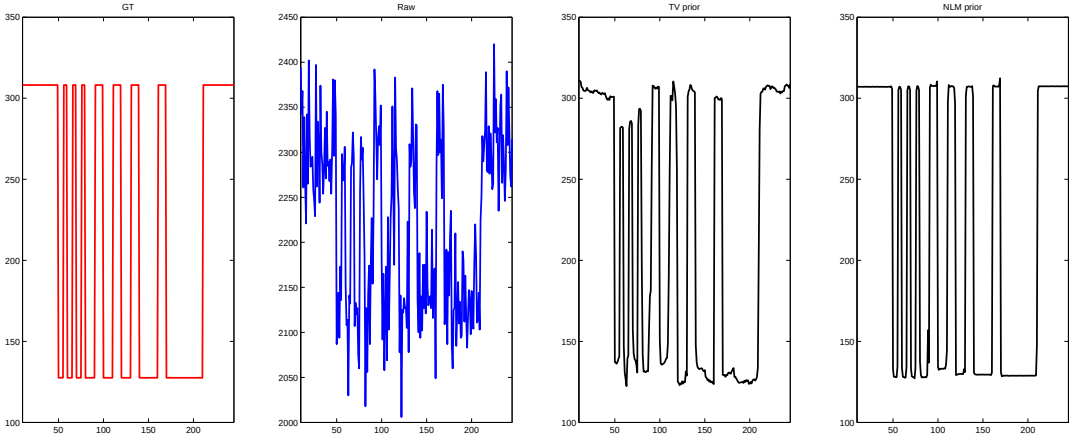
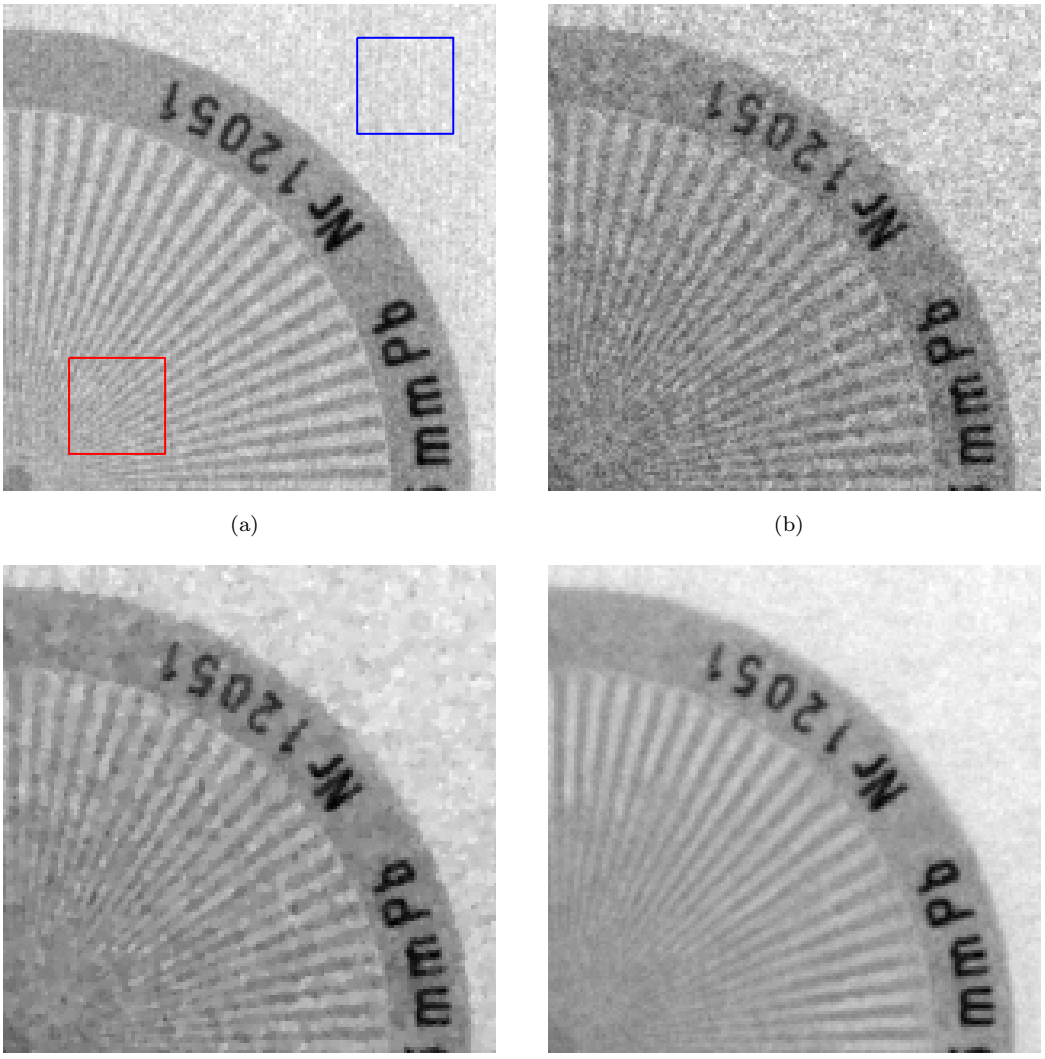
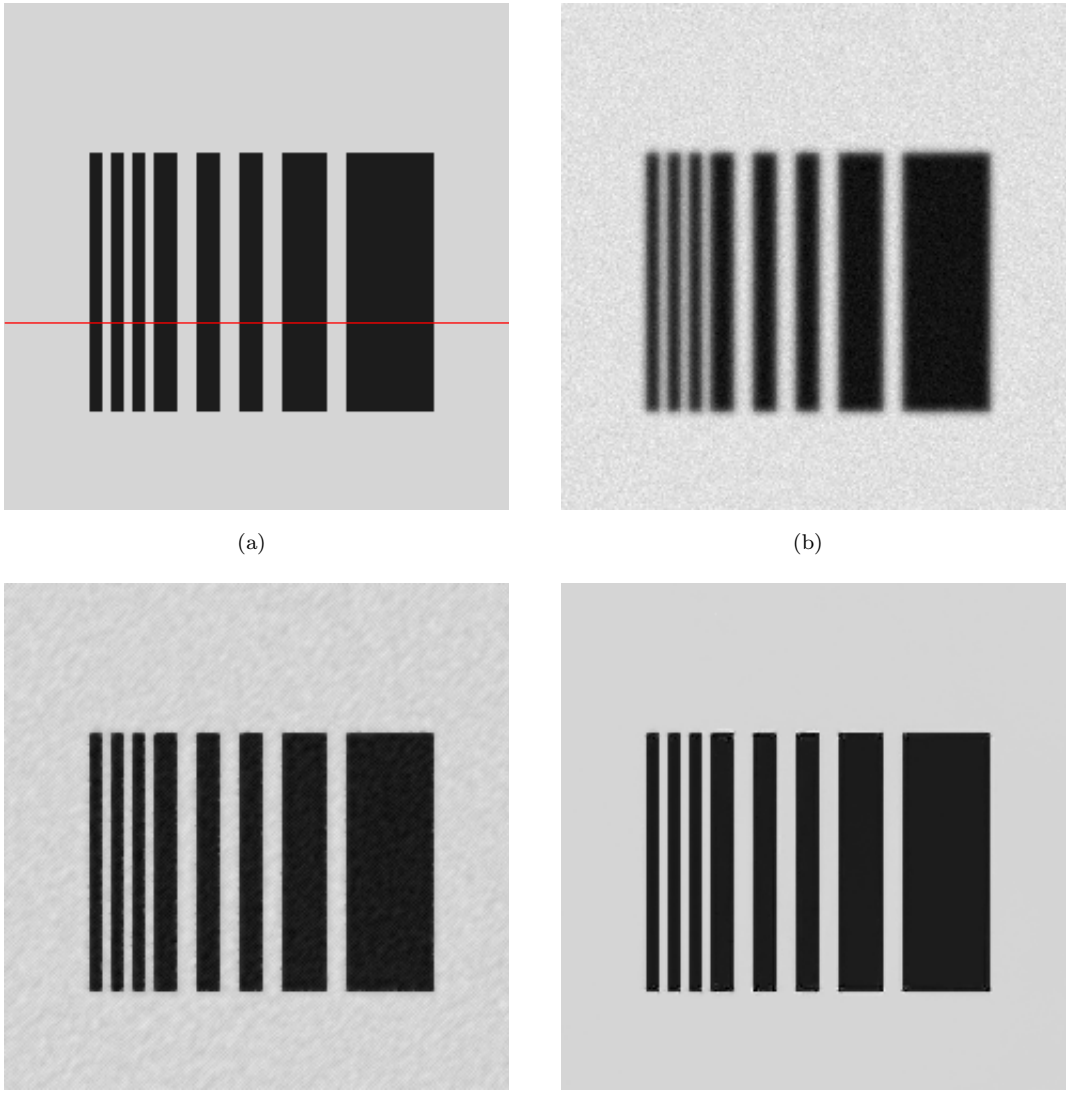


FIGURE 6.2: Profile from a horizontal line (location indicated by the red line in (a)) for the images in Figure 6.1 (a) to (d)



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

FIGURE 6.3: Scatter correction on a phantom image. Scatter model: the constant scatter level. The contrast is calculated using the method in [64, 65]: the blue line and red line windows indicate, respectively, the locations for "background" and "signal" for this calculation. (a)clinical ground truth, contrast=6.68%; (b)scattered image, contrast=3.32%; (c)result from TV, contrast=9.10%; (d)result from our NLM algorithm, contrast=9.51%.



(a) (b)

(c) (d)

FIGURE 6.4: Scatter correction on a simulated image. Scatter model: nonlinear convolution. (a)ground truth; (b)scattered image; (c)result from TV, PSNR=27.65dB; (d)result from our NLM algorithm, PSNR=36.78dB.

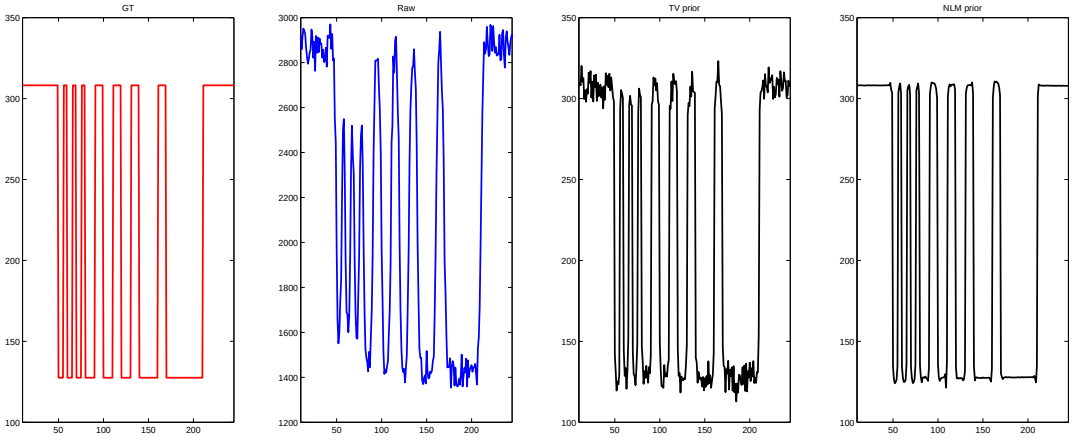
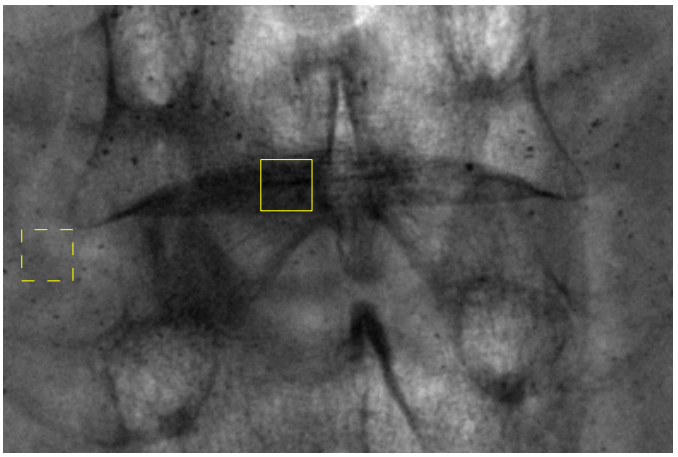
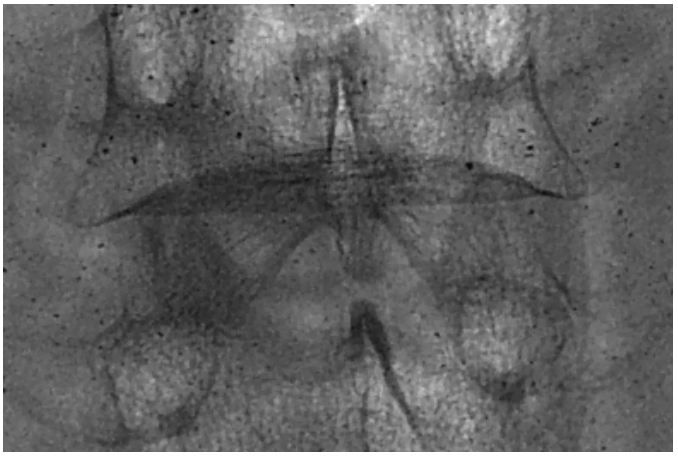


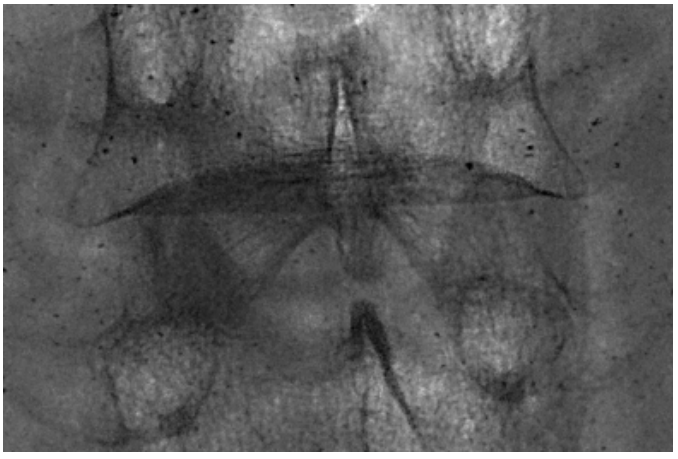
FIGURE 6.5: Profile from a horizontal line (location indicated by the red line in (a)) for the images in Figure 6.4 (a) to (d)



(a)

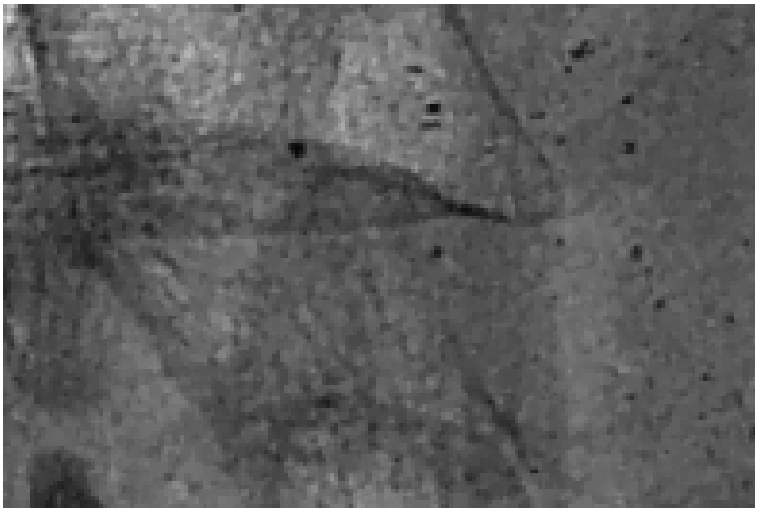


(b)

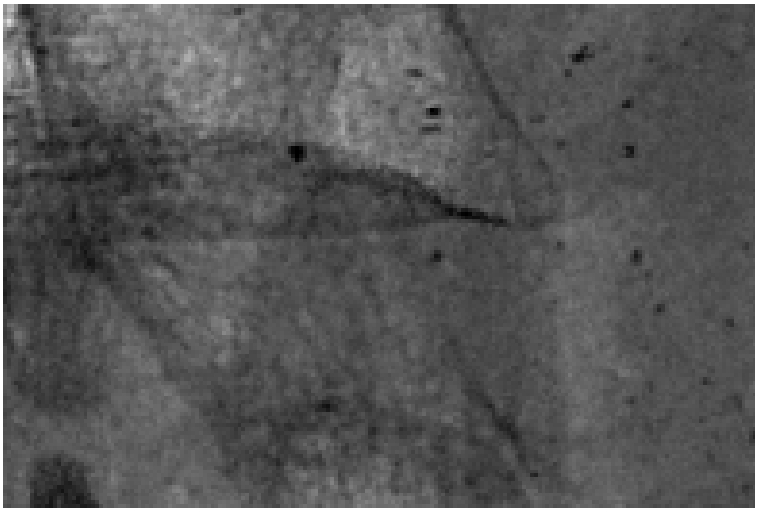


(c)

FIGURE 6.6: Scatter correction on an abdomen phantom image. Scatter model: nonlinear convolution. The contrast is calculated using the method in [64, 65]: the dashed line and solid line windows indicate, respectively, the locations for "background" and "signal" for this calculation. (a)input image, contrast=10.70%; (b)result from TV, contrast=12.56%; (c)result from our NLM algorithm, contrast=12.92%.



(a)



(b)

FIGURE 6.7: Scatter correction on an abdomen phantom image. (d)a zoomed-in region of (b); (e)a zoomed-in region of (c).

References

[1] S. Lloyd, Least squares quantization in PCM. *IEEE transactions on Information Theory*, 28(2), pp. 129-137, 1982.

[2] R. Sibson, SLINK: an optimally efficient algorithm for the single-link cluster method. *The Computer Journal (British Computer Society)*, 16(1), pp. 30-34, 1973.

[3] D. Defays, An efficient algorithm for a complete link method. *The Computer Journal (British Computer Society)*, 20(4), pp. 364-366, 1977.

[4] C. M. Bishop, Pattern Recognition and Machine Learning. *Springer*, 2007.

[5] K. P. Murphy, Machine Learning: A Probabilistic Perspective. *the MIT Press*, 2012.

[6] M. Sonka, V. Hlavac, and R. Boyle, Image Processing, Analysis, and Machine Vision. *Cengage Learning; 3 edition*, 2008.

[7] H. Kriegel, P. Kroger, J. Sander, and A. Zimek, Density-based clustering. *WIREs Data Mining and Knowledge*, 1(3), pp. 231-240, 2011.

[8] A. Rodriguez and A. Laio, Clustering by fast search and find of density peaks. *Science*, 344, pp. 1492-1496, 2014.

[9] M. Verleysen and D. Francois, The curse of dimensionality in data mining and time series prediction. *IWANN05 Proceedings of the 8th international conference on Artificial Neural Networks: computational Intelligence and Bioinspired Systems*, pp. 758–770, 2005.

[10] I. J. Myung, Tutorial on maximum likelihood estimation. *Journal of Mathematical Psychology*, 47, pp. 90-100, 2003.

[11] S. Purcell, Maximum Likelihood Estimation.

[12] J. Gauvain and C. Lee, Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains. *IEEE Trans. On Speech and Audio Processing*, pp. 291-298, 1994.

[13] A. P. Dempster, N. M. Laird and D.B. Rubin, Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society*, Series B,39(1), pp. 1-38, 1977.

[14] Z. Ghahramani, and M. I. Jordan, Supervised learning from incomplete data via an EM approach. *Advances in Neural information Processing Systems*, 6, pp. 120-127, 1994.

[15] L. Xu and M. I. Jordan, On Convergence Properties of the EM Algorithm for Gaussian Mixtures. *Neural Computation*, 8, pp. 129-151, 1995.

[16] J. Bilmes, A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models. 1998.

[17] R.M. Neal, Markov chain sampling methods for Dirichlet process mixture models. *Journal of Computational and Graphical Statistics*, 9(2), pp. 249-265, 2000.

[18] D. Gorur and C. E. Rasmussen, Dirichlet process Gaussian mixture models: choice of base distribution. *Journal of Computer Science and Technology*, 25(4), pp. 615-626, 2010.

[19] F. Nielsen and V. Garcia, Statistical exponential families: A digest with flash cards. 2011

[20] K. P. Murphy, Conjugate Bayesian analysis of the Gaussian distribution. 2007

[21] N. Ueda and R. Nakano, Deterministic annealing EM algorithm. *Neural Networks*, 11(2), pp. 271-282, 1998.

[22] G. Elidan, M. Ninio, N. Friedman and D. Schuurmans, Data perturbation for escaping local maxima in learning. *Eighteenth national conference on Artificial Intelligence*, pp. 132-139, 2002.

[23] Z. Zhang, B. T. Dai, and A. K. H. Tung, Estimating local optimums in EM algorithm over Gaussian mixture model. *ICML 08 Proceedings of the 25th international conference on Machine Learning*, pp. 1240-1247, 2008.

[24] N. Vlassis and A. Likas, A Greedy EM Algorithm for Gaussian Mixture Learning. *Neural Processing Letters*, 15, pp. 77-87, 2002.

[25] J. J. Verbeek, N. Vlassis and B. Krose, Efficient Greedy Learning of Gaussian Mixture Models. *Neural Computation*, 15(2), pp. 469-485, 2003.

[26] J. Zhang, The Mean Field Theory in EM Procedures for Markov Random Fields, *IEEE Transactions on Signal Processing*, 40(10), pp. 2570-2583, 1992.

[27] A Wilcox, K Natarajan, C Weng, Using personal health records for automated clinical trials recruitment: the ePaIRing model. *Proc. AMIA Translational Bioinformatics Summit 2009*, pp. 136-140, 2009

[28] J. Kamal et al, Using an information warehouse to screen patients for clinical trials: a prototype. *Proc. 2005 AMIA Annual Symp*, pp. 1004, 2005.

[29] Lonsdale et al., Assessing clinical trial eligibility with logic expression queries. *Data & Knowledge Engineering*, 66(1), pp. 3-17, 2008.

[30] L. Rokach et al., Information retrieval system for medical narrative reports. *Lecture Notes in Computer Science,Springer*, 3055, pp. 217-228, 2004.

[31] B. MacCartney et al., Learning to recognize features of valid textual entailment. *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL 2006)*, 2006.

[32] B. MacCartney and C. Manning, Modeling semantic containment and exclusion in natural language inference. *The 22nd International Conference on Computational Linguistics (Coling-08), Manchester, UK*, 2008.

[33] A. Hickl et al., Recognizing textural entailment with LCC's Groundhog System. *Proc. 2nd PASCAL Challenges Workshop on Recognizing Textural Entailment*, 2006.

[34] A. Hickl and J. Bensley, A discourse commitment-based framework for recognizing textual entailment. *Proc. ACL*, 2007.

[35] L. Li, H. S. Chase, C. O. Patel, C. Friedman, and C. Weng, Comparing ICD9-encoded diagnoses and NLP-processed discharge summaries for clinical trials pre-Screening: A case study. *AMIA Annu Symp Proc. 2008*, pp. 404-408, 2008.

[36] D. Jurafsky and J. J. Martin, Speech and Language Processing, *Prentice-Hall, 2nd Edition*, 2008.

[37] Stanford software: parsing and subtree search. *http://nlp.stanford.edu/software/lex-parser.shtml, and http://nlp.stanford.edu/software/tregex.shtml*

[38] A reference about WordNet, *http://search.cpan.org/dist/WordNet-Similarity/*

[39] A reference about PubMed website. *http://www.ncbi.nlm.nih.gov/*

[40] J. Zhang, Y. Gu, etc., Automatic patient search for breast cancer clinical trials using free-text medical reports. *ACM International Health Informatics Symposium, IHI 2010*, pp. 405-409, 2010.

[41] Y. Gu, C. Kallas, J. Zhang, etc., Automatic Patient Search Using Bernoulli Model. *Healthcare Informatics (ICHI) 2013 IEEE International Conference on*, pp. 517-522, 2013.

[42] C. Manning, P. Raghavan, and H. Schutze, Introduction to Information Retrieval. *Cambridge University Press*, 2008.

[43] D. Losada, Language modeling for sentence retrieval: A comparison between multiple-Bernoulli models and multinomial models. *Information Retrieval and Theory Workshop, Glasgow, UK*, 2005.

[44] D. Losada and L. Azzopardi, Assessing Multivariate Bernoulli Models for Information Retrieval. *ACM Transactions on Information Systems*, 26(3), 2008.

[45] R. A. Aronson, Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program. *Proceedings of 2001 AMIA Symposium*, pp. 17-21, 2001.

[46] S. Deerwester et al., Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6), pp. 391-407, 1990.

[47] T. Mikovlov, et al., Distributed Representation of Words and Phrases and their Compositionality. In Advances on Neural Information Processing Systems, 2013c

[48] Q. Le and T. Mikovlov, Distributed Representation of Sentences and Documents. ICML 2014

[49] http://radimrehurek.com/gensim/models/doc2vec.html

[50] https://dumps.wikimedia.org/enwiki/

[51] Y. Saeys, I. Inza and P. Larranaga, A review of feature selection techniques in bioinformatics. *Bioinformatics*, Vol. 23 no. 19, pp. 2507-2517, 2007.

[52] S. Beniwal and J. Arora, Classification and Feature Selection Techniques in Data Mining. *International Journal of Engineering Research and Technology(IJERT)*, Vol. 1 Issue 6, 2012.

[53] R. Kohavi, A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection, *International Joint Conference on Artificial Intelligence(IJCAI)*, 1995.

[54] Pedregosa et al., Scikit-learn: Machine Learning in Python, *Journal of Machine Learning Research* Vol. 12, pp. 2825-2830, 2011.

[55] L. Breiman, Random Forests, *Machine Learning*, 45(1), 5-32, 2001.

[56] E.-P. Ruhrnshopf and K. Klingenbeck, A general framework and review of scatter correction methods in x-ray cone beam computerized tomography. Part 1: scatter compensation approaches. *Med. Phys.*, 38(7), pp.4296-4311, 2011.

[57] E.-P. Ruhrnshopf and K. Klingenbeck, A general framework and review of scatter correction methods in x-ray cone beam computerized tomography. Part 2: scatter estimation approaches. *Med. Phys.*, 38(9), pp.5186-5199, 2011.

[58] A. Buades, B. Coll, and J. M. Morel, A review of image denoising algorithms, with a new one. *Multiscale Model. Simul.*, 4, pp.490-530, 2005.

[59] C. A. Bouman and K. Saue, A unified approach to statistical tomography using coordinate descent optimization. *IEEE Trans. Image Processing*, 5(3), pp.480-492, 1996.

[60] X. Zhang, et al., Bregmanized nonlocal regularization for deconvolution and sparse reconstruction. *SIAM J. Imaging Sciences*, 3(3), pp.253-276, 2010.

[61] B. De Man, Iterative reconstruction for reduction of metal artifacts in comptued tomography. *PhD thesis*, 2001.

[62] A. Levin et al., Efficient marginal likelihood optimization in blind deconvolution. *2011 IEEE CVPR Conf.*, pp. 2657 - 2664, 2011.

[63] D. Perrone and P. Favaro, A clearer picture of total variation blind deconvolution. *IEEE PAMI*, 38(6), 2016.

[64] L. Shi et al., Library-based scatter correction for dedicated cone beam breast CT: a feasibility study. *Medical Imaging 2016: Physics of Medical Imaging, Proc. SPIE*, 9783, pp.978330-1 to 978330-6, 2016.

[65] R. Rana, et al., Scatter estimation and removal of anti-scatter grid-line artifacts from anthropomorphic head phantom images taken with a high resolution image detector. *Medical Imaging 2016: Physics of Medical Imaging, Proc. SPIE*, 9783, pp.978364-1 to 978364-10, 2016.

Appendix A: Derivation of EM GMM

$$\boldsymbol{\theta}^{(t+1)} = \arg\max_{\boldsymbol{\theta}} \left\{ Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) \right\} \tag{6.16}$$

$$\nabla_{\boldsymbol{\theta}} \left\{ Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) \right\} = 0 \tag{6.17}$$

$$\nabla_{\boldsymbol{\theta}} \left\{ \sum_i \sum_j E[z_{i,j}|y_i, \boldsymbol{\theta}^{(t)}] \log p(y_i|z_j, \boldsymbol{\theta}^{(t)}) + \sum_i \sum_j E[z_{i,j}|y_i, \boldsymbol{\theta}^{(t)}] \log \pi_j \right\} = 0 \tag{6.18}$$

$$\nabla_{\boldsymbol{\theta}} \left\{ \sum_i \sum_j E[z_{i,j}|y_i, \boldsymbol{\theta}^{(t)}] \log \left(\frac{1}{(2\pi)^{d/2}|\Sigma_j|^{1/2}} \exp \left\{ -\frac{1}{2}(y_i - \mu_j)^T (\Sigma_j)^{-1} (y_i - \mu_j) \right\} \right) + \sum_i \sum_j E[z_{i,j}|y_i, \boldsymbol{\theta}^{(t)}] \log \pi_j \right\} = 0 \tag{6.19}$$

Estimation of $\hat{\boldsymbol{\mu}}$

$$\nabla_{\boldsymbol{\mu}} \left\{ Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) \right\} = 0 \tag{6.20}$$

$$\nabla_{\mu_j}\left\{\sum_i\sum_j E[z_{i,j}|y_i,\boldsymbol{\theta}^{(t)}]\log\left(\frac{1}{(2\pi)^{d/2}|\Sigma_j|^{1/2}}\exp\left\{-\frac{1}{2}(y_i-\mu_j)^T(\Sigma_j)^{-1}(y_i-\mu_j)\right\}\right)\right\}=0 \tag{6.21}$$

$$\sum_i E[z_{i,j}|y_i,\boldsymbol{\theta}^{(t)}]\frac{y_i-\mu_j}{\Sigma_j}=0 \tag{6.22}$$

$$\hat{\mu}_j^{(t+1)}=\frac{\sum_{i=1}^n E[z_{i,j}|y_i,\boldsymbol{\theta}^{(t)}]y_i}{\sum_{i=1}^n E[z_{i,j}|y_i,\boldsymbol{\theta}^{(t)}]} \tag{6.23}$$

Estimation of $\hat{\boldsymbol{\Sigma}}$

$$\nabla_{\boldsymbol{\Sigma}}\left\{Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)})\right\}=0 \tag{6.24}$$

$$\nabla_{\Sigma_j}\left\{\sum_i\sum_j E[z_{i,j}|y_i,\boldsymbol{\theta}^{(t)}]\log\left(\frac{1}{(2\pi)^{d/2}|\Sigma_j|^{1/2}}\exp\left\{-\frac{1}{2}(y_i-\mu_j)^T(\Sigma_j)^{-1}(y_i-\mu_j)\right\}\right)\right\}=0 \tag{6.25}$$

$$-\frac{1}{2}\sum_i E[z_{i,j}|y_i,\boldsymbol{\theta}^{(t)}]\frac{1}{|\Sigma_j|}|\Sigma_j|(\Sigma_j)^{-1}+\frac{1}{2}\sum_i E[z_{i,j}|y_i,\boldsymbol{\theta}^{(t)}](\Sigma_j)^{-1}(y_i-\mu_j)^T(y_i-\mu_j)(\Sigma_j)^{-1}=0 \tag{6.26}$$

$$\hat{\Sigma}_j^{(t+1)}=\frac{\sum_{i=1}^n E[z_{i,j}|y_i,\boldsymbol{\theta}^{(t)}](y_i-\hat{\mu}_j^{(t+1)})^T(y_i-\hat{\mu}_j^{(t+1)})}{\sum_{i=1}^n E[z_{i,j}|y_i,\boldsymbol{\theta}^{(t)}]} \tag{6.27}$$

Estimation of $\hat{\boldsymbol{\pi}}$

$$\nabla_{\boldsymbol{\pi}} \left\{ Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) \right\} = 0 \tag{6.28}$$

$$\nabla_{\boldsymbol{\pi}} \left\{ \sum_i \sum_j E[z_{i,j}|y_i, \boldsymbol{\theta}^{(t)}] \log \pi_j \right\} = 0 \tag{6.29}$$

Given the equation of $\sum_{j=1}^{K} \pi_j = 1$,

$$\Lambda = \sum_i \sum_j E[z_{i,j}|y_i, \boldsymbol{\theta}^{(t)}] \log \pi_j + \lambda \left(\sum_j \pi_j - 1 \right) \tag{6.30}$$

$$\nabla_{\boldsymbol{\pi}} \{\Lambda\} = 0 \tag{6.31}$$

$$\begin{aligned} \sum_i E[z_{i,j}|y_i, \boldsymbol{\theta}^{(t)}] * \frac{1}{\pi_j} + \lambda &= 0 \\ \sum_i E[z_{i,j}|y_i, \boldsymbol{\theta}^{(t)}] * \frac{1}{\pi_j} &= -\lambda \pi_j \\ \sum_j \sum_i E[z_{i,j}|y_i, \boldsymbol{\theta}^{(t)}] * \frac{1}{\pi_j} &= \sum_j (-\lambda \pi_j) \\ n &= -\lambda \end{aligned}$$

$$\hat{\pi}_j^{(t+1)} = \frac{\sum_{i=1}^{n} E[z_{i,j}|y_i, \boldsymbol{\theta}^{(t)}]}{n} \tag{6.32}$$

Appendix B: Derivation of MAP EM GMM

$$
\begin{aligned}
Q'(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) = &\sum_i \sum_j r_{i,j} \log \left(\frac{1}{(2\pi)^{d/2}|\Sigma_j|^{1/2}} \exp\left\{ -\frac{1}{2}(y_i - \mu_j)^T (\Sigma_j)^{-1} (y_i - \mu_j) \right\} \right) \\
&+ \sum_i \sum_j r_{i,j} \log \pi_j + \sum_j \log \left(\frac{1}{B(\boldsymbol{\alpha})} \right) + \sum_j (\alpha_i - 1) \log \pi_j \\
&+ \sum_j \log \left(\frac{1}{(2\pi)^{d/2} \left| \frac{1}{\beta_0} \Sigma_j \right|^{1/2}} \exp\left\{ -\frac{1}{2}(\mu_j - m_0)^T \left(\frac{1}{\beta_0} \Sigma_j \right)^{-1} (\mu_j - m_0) \right\} \right) \\
&+ \sum_j \log \left(\frac{|S_0|^{\nu_0/2}}{2^{\frac{\nu_0 d}{2}} \Gamma_d(\frac{\nu_0}{2})} |\Sigma_j|^{-(\nu_0+d+1)/2} \exp\left\{ -\frac{1}{2} Tr(S_0 {\Sigma_j}^{-1}) \right\} \right) \qquad (6.33)
\end{aligned}
$$

$$
\boldsymbol{\theta}^{(t+1)} = \arg\max_{\boldsymbol{\theta}} \left\{ Q'(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) \right\} \qquad (6.34)
$$

$$
\nabla_{\boldsymbol{\theta}} \left\{ Q'(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) \right\} = 0 \qquad (6.35)
$$

Estimation of $\hat{\boldsymbol{\mu}}'$

$$
\nabla_{\boldsymbol{\mu}} \left\{ Q'(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) \right\} = 0 \qquad (6.36)
$$

$$\nabla_{\mu_j}\left\{\sum_i\sum_j r_{i,j}\log\left(\frac{1}{(2\pi)^{d/2}|\Sigma_j|^{1/2}}\exp\left\{-\frac{1}{2}(y_i-\mu_j)^T(\Sigma_j)^{-1}(y_i-\mu_j)\right\}\right)\right.$$
$$\left.+\sum_j\log\left(\frac{1}{(2\pi)^{d/2}\left|\frac{1}{\beta_0}\Sigma_j\right|^{1/2}}\exp\left\{-\frac{1}{2}(\mu_j-m_0)^T\left(\frac{1}{\beta_0}\Sigma_j\right)^{-1}(\mu_j-m_0)\right\}\right)\right\}=0 \quad (6.37)$$

$$\sum_i r_{i,j}\frac{y_i-\mu_j}{\Sigma_j}+\frac{\beta_0(\mu_j-m_0)}{\Sigma_j}=0 \quad (6.38)$$

$$\hat{\mu}_j^{(t+1)}=\frac{\sum_{i=1}^n r_{i,j}y_i+\beta_0 m_0}{\sum_{i=1}^n r_{i,j}+\beta_0} \quad (6.39)$$

Estimation of $\hat{\mathbf{\Sigma}}'$

$$\nabla_{\mathbf{\Sigma}}\left\{Q'(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)})\right\}=0 \quad (6.40)$$

$$\nabla_{\Sigma_j}\left\{\sum_i\sum_j r_{i,j}\log\left(\frac{1}{(2\pi)^{d/2}|\Sigma_j|^{1/2}}\exp\left\{-\frac{1}{2}(y_i-\mu_j)^T(\Sigma_j)^{-1}(y_i-\mu_j)\right\}\right)\right.$$
$$+\sum_j\log\left(\frac{1}{(2\pi)^{d/2}\left|\frac{1}{\beta_0}\Sigma_j\right|^{1/2}}\exp\left\{-\frac{1}{2}(\mu_j-m_0)^T\left(\frac{1}{\beta_0}\Sigma_j\right)^{-1}(\mu_j-m_0)\right\}\right)$$
$$\left.+\sum_j\log\left(\frac{|S_0|^{\nu_0/2}}{2^{\frac{\nu_0 d}{2}}\Gamma_d(\frac{\nu_0}{2})}|\Sigma_j|^{-(\nu_0+d+1)/2}\exp\left\{-\frac{1}{2}Tr(S_0{\Sigma_j}^{-1})\right\}\right)\right\}=0 \quad (6.41)$$

$$
\begin{aligned}
&-\frac{1}{2}\sum_i r_{i,j}\frac{1}{|\Sigma_j|}|\Sigma_j|(\Sigma_j)^{-1}+\frac{1}{2}\sum_i r_{i,j}(\Sigma_j)^{-1}(y_i-\mu_j)^T(y_i-\mu_j)(\Sigma_j)^{-1}\\
&-\frac{1}{2}\frac{1}{\beta_0}\frac{1}{\left|\frac{1}{\beta_0}\Sigma_j\right|}\left|\frac{1}{\beta_0}\Sigma_j\right|\left(\frac{1}{\beta_0}\Sigma_j\right)^{-1}+\frac{1}{2}\frac{1}{\beta_0}\left(\frac{1}{\beta_0}\Sigma_j\right)^{-1}(\mu_j-m_0)^T(\mu_j-m_0)\left(\frac{1}{\beta_0}\Sigma_j\right)^{-1}\\
&-\frac{\nu_0+d+1}{2}\frac{1}{|\Sigma_j|}|\Sigma_j|(\Sigma_j)^{-1}+\frac{1}{2}(\Sigma_j)^{-1}S_0(\Sigma_j)^{-1}=0 \quad (6.42)
\end{aligned}
$$

After simplifying the above equation, we can get

$$
\hat{\Sigma}_j^{(t+1)}=\frac{S_0+\sum_{i=1}^n r_{i,j}\left(y_i-\hat{\mu}_j^{(t+1)}\right)^T\left(y_i-\hat{\mu}_j^{(t+1)}\right)+\beta_0\left(\hat{\mu}_j^{(t+1)}-m_0\right)^T\left(\hat{\mu}_j^{(t+1)}-m_0\right)}{\sum_{i=1}^n r_{i,j}+\beta_0+\nu_0+D+2} \tag{6.43}
$$

Estimation of $\hat{\boldsymbol{\pi}}'$

$$
\nabla_{\boldsymbol{\pi}}\left\{Q'(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)})\right\}=0 \tag{6.44}
$$

$$
\nabla_{\boldsymbol{\pi}}\left\{\sum_i\sum_j r_{i,j}\log\pi_j+\sum_j(\alpha_i-1)\log\pi_j\right\}=0 \tag{6.45}
$$

Given the equation of $\sum_{j=1}^K \pi_j=1$,

$$
\Lambda'=\sum_i\sum_j r_{i,j}\log\pi_j+\sum_j(\alpha_i-1)\log\pi_j+\lambda'\left(\sum_j\pi_j-1\right) \tag{6.46}
$$

$$
\nabla_{\boldsymbol{\pi}}\{\Lambda'\}=0 \tag{6.47}
$$

$$\sum_i r_{i,j} * \frac{1}{\pi_j} + \alpha_j * \frac{1}{\pi_j} - \frac{1}{\pi_j} + \lambda' = 0$$

$$\sum_i r_{i,j} + (\alpha_j - 1) = -\lambda' \pi_j$$

$$\sum_j \sum_i r_{i,j} + \sum_j (\alpha_j - 1) = \sum_j (-\lambda' \pi_j)$$

$$n + \sum_j \alpha_j - K = -\lambda'$$

$$\hat{\pi}_j^{(t+1)} = \frac{\sum_{i=1}^{n} r_{i,j} + \alpha_j - 1}{n + \sum_{j=1}^{K} \alpha_j - K} \tag{6.48}$$

Curriculum Vitae

Yingying Gu

Place of birth: Tianjin, China

Education:

- B.A., Beijing Institute of Petrochemical Technology, May 2003
  Major: Chemical Engineering
- M.S., China University of Mining & Technology, December 2006
  Major: Computer Science
- M.S., Marquette University, August 2011
  Major: Computational Chemistry

Dissertation Title: Bayesian Methods and Machine Learning for Processing Text and Image Data

Publications:

- Scatter Reduction by Non-local Techniques, SPIE Medical Imaging, 2017.
- Automatic Patient Search Using Bernoulli Model, International Workshop on Data Mining for Healthcare, ICHI, 2013.
- Automatic Patient Search for Breast Cancer Clinical Trials Using Free-Text Medical Report, 1st ACM International Health Informatics Symposium, 2010.
- Patient Information Search for Breast Cancer Clinical Trials, AMIA 2010 Annual Symposium, 2010.
- On the role of vibrational anharmonicities in a two-qubit system, Journal of Chemical Physics, 2009.
- A new approach to compute the Euler Number of 3D image, Industrial Electronics and Applications, 2008. ICIEA 2008.

- Calculation of the specific surface area of the froth in flotation, Journal of China Coal Society, 2007.
- Feature extraction based on image segmentation of coal flotation froth, Journal of China Coal Society, 2007.
- A Classification of Flotation Froth Based on Geometry, Mechatronics and Automation, ICMA 2007.
- The Euler Number Expressed as Neighbor Number and Application, IMECS 2007.