Large-scale Consensus Clustering and Data Ownership Considerations for Medical Applications

by
Chidube Donald Ezeozue
B.Eng., University of Nigeria, Nsukka (2008)

Submitted to the Engineering Systems Division
and
Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degrees of
Master of Science in Technology and Policy
and
Master of Science in Electrical Engineering and Computer Science
at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
September 2013

Author . . . .
Technology and Policy Program, Engineering Systems Division
Department of Electrical Engineering and Computer Science
July 5, 2013

Certified by . . . .
Una-May O'Reilly
Principal Research Scientist, CSAIL
Thesis Supervisor

Certified by . . . .
Kalyan Veeramachaneni
Research Scientist, CSAIL
Thesis Supervisor

Accepted by . . . .
Professor Dava J. Newman
Director, Technology and Policy Program

Accepted by . . . .
Professor Leslie A. Kolodziejski
Chair, Committee on Graduate Students
Department of Electrical Engineering and Computer Science

Large-scale Consensus Clustering and Data Ownership Considerations for Medical Applications

by

Chidube Donald Ezeozue

Submitted to the Engineering Systems Division

and

Department of Electrical Engineering and Computer Science
on July 5, 2013, in partial fulfillment of the
requirements for the degrees of
Master of Science in Technology and Policy

and

Master of Science in Electrical Engineering and Computer Science

Abstract

An intersection of events has led to a massive increase in the amount of medical data being collected from patients inside and outside the hospital. These events include the development of new sensors, the continuous decrease in the cost of data storage, the development of *Big Data* algorithms in other domains and the Health Information Technology for Economic and Clinical Health (HITECH) Act's $20 billion incentive for hospitals to install and use Electronic Health Record (EHR) systems. The data being collected presents an excellent opportunity to improve patient care.

However, this opportunity is not without its challenges. Some of the challenges are technical in nature, not the least of which is how to efficiently process such massive amounts of data. At the other end of the spectrum, there are policy questions that deal with data privacy, confidentiality and ownership to ensure that research continues unhindered while preserving the rights and interests of the stakeholders involved.

This thesis addresses both ends of the challenge spectrum. First of all, we design and implement a number of methods for automatically discovering groups within large amounts of data, otherwise known as clustering. We believe this technique would prove particularly useful in identifying patient states, segregating cohorts of patients and hypothesis generation. Specifically, we scale a popular clustering algorithm, Expectation-Maximization (EM) for Gaussian Mixture Models to be able to run on a cloud of computers. We also give a lot of attention to the idea of Consensus Clustering which allows multiple clusterings to be merged into a single ensemble clustering. Here, we scale one existing consensus clustering algorithm, which relies on

EM for multinomial mixture models. We also develop and implement a more general framework for retrofitting any consensus clustering algorithm and making it amenable to streaming data as well as distribution on a cloud.

On the policy end of the spectrum, we argue that the issue of data ownership is essential and highlight how the law in the United States has handled this issue in the past several decades, focusing on common law and state law approaches. We proceed to identify the flaws, especially the fragmentation, in the current system and make recommendations for a more equitable and efficient policy stance. The recommendations center on codifying the policy stance in Federal Law and allocating the property rights of the data to both the healthcare provider and the patient.

Thesis Supervisor: Una-May O'Reilly
Title: Principal Research Scientist, CSAIL

Thesis Supervisor: Kalyan Veeramachaneni
Title: Research Scientist, CSAIL

Acknowledgments

This thesis would not have been possible without the guidance, encouragement and funding from my advisor, Una-May O'Reilly. I am especially grateful for the latitude at the beginning of my time in her research group to explore a wide range of topics and ideas which we gradually narrowed down to form this work. I am also grateful for Kalyan Veeramachaneni's guidance and ideas throughout my time in the group as well as his painstaking review of this thesis.

My interest in the subject of medical data ownership was triggered by a tour of the Beth Isreal Deaconess Medical Center given to our research group by Leo Celi. This interest was nurtured and further stimulated by later conversations with him and a number of stakeholders he set me up to speak with and I am more than grateful for that.

My time here was partly funded by a generous fellowship from the Legatum Center for Development and Entrepreneurship. The Center was also very instrumental in getting me started on my entrepreneurial journey through the mentorship, classes and exposure provided. I look forward to seeing where this journey will lead.

My 2-year journey through MIT has been eventful especially because of the people I have interacted with here, from my lab mates to classmates and staff in the Technology and Policy Program. I have also been fortunate to have awesome and supportive room mates: Abhijit, Cole, Deniz and Esther. My cousin and friend, Ekene, who might have as well been a room mate given the amount of time I spent in his room made the last one year of being here a lot less lonely. I also met some fantastic people in my bible study group who both inspired me by their character and lives as well as provided much-needed support when we prayed and shared together.

Finally, I am grateful for my parents and siblings, Onyi, Eky and Zim who have been extremely supportive throughout these two years in more ways than I can mention. Thank you for believing in me as much as you do.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This thesis focuses on technical and policy challenges in knowledge mining of medical data. In this chapter, we describe the motivation for the choice of problem space and methods. We also give an overview of the main contributions of this work.

1.1 Motivation

We are at an important point in history that necessitates a focus on what may be called *Big Medical Data*. Due to multiple drivers, there has been an explosion in the amount of medical data collected by several means for different purposes. The first of these drivers is the cheap availability of data storage. Figure 1-1 shows the dramatic decline in the cost of data storage over the last two decades. The collection of data in the United States is also accelerated by the Federal Government's $20 billion incentive to adopt electronic health records. This incentive is contained within the Health Information Technology for Economic and Clinical Health (HITECH) Act which stipulates, among other things, that healthcare providers be paid significant sums of money if they can attest to meaningful use of an electronic health record (EHR) system (Blumenthal, 2010). Finally, an increase in personal monitoring (Milenković et al., 2006) has equally led to increased gathering of personal physiological data. Personal monitoring includes portable Holter monitors for taking ECG measurements on the move and other wearable devices like FitBit[1] and Nike's FuelBand[2] which monitor human activity such as sleep, steps walked, calories burned, etc.

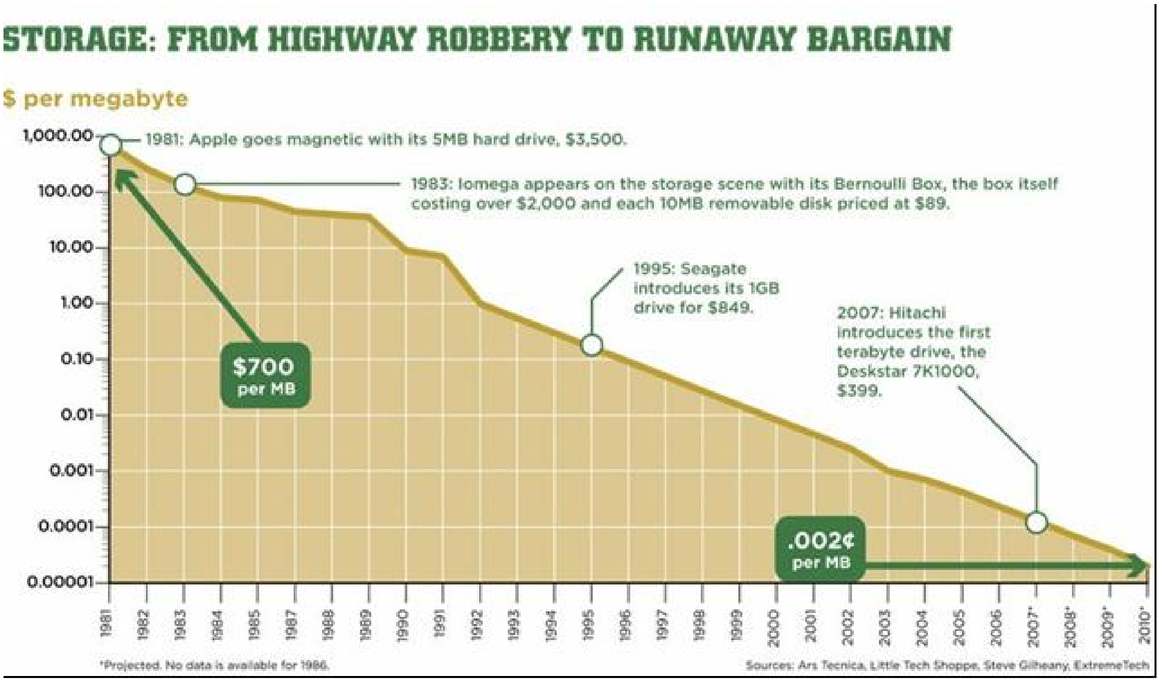


Figure 1-1: Plummeting data storage costs (Smith and Williams, 2010)

While the increase in data collection is commendable, we fear its utility might be severely limited by a human's limited ability to process large amounts of data that typically spans several hours/days of historical information, may include multiple data streams from multiple indices measured and is gathered from multiple individuals. This volume, variety and velocity of data therefore calls for scalable and innovative knowledge mining technology such as machine learning which has been applied in other *Big Data* domains. For instance, large amounts of data generated by online services like Google, Amazon, Twitter as well as brick-and-mortar businesses like Walmart has led to methods that can mine such data and has resulted in advances in collaborative filtering, recommendation systems and smart product placement in stores thereby leading to drastic improvements in the way these businesses operate.

[1] www.fitbit.com

[2] www.nike.com/us/en_us/c/nikeplus-fuelband

There is a similar opportunity to improve and lower the cost of patient care using medical knowledge mining. Given the frequency of errors (Schoen et al., 2005) and high costs (Brill, 2013) associated with health care, there is an opportunity to optimize hospital resources by allocating more resources to sicker patients and supporting doctor decision-making by automatically extracting patterns from a patient's history and other similar patients.

This thesis is both a technology and a policy thesis centered around the opportunity to improve patient care using knowledge mining of medical data. On the technology end, we focus on a specific method of knowledge mining known as clustering. This method, which falls within the larger domain of *unsupervised machine learning*, enables the discovery of groups within data which, in contrast to *supervised learning*, does not have to be labelled by a domain expert or data for which the labels may not even be known. The clustering method is particularly suited to medical data mining because getting such labelled datasets may be expensive and time-consuming given the volume of data in question.

On the policy end, we shift focus from the methods for analyzing the data to the policy stances that ensure the availability of data. We are specifically interested in the issue of data ownership and the policy directions that will ensure the availability of large amounts of medical data for knowledge mining and consequent improvements in the quality of patient care. The next two sections will briefly discuss the contributions this thesis makes to both large-scale clustering and the subject of medical data ownership.

1.2 Clustering

Clustering is an unsupervised learning method for discovering hidden patterns in data without any reference to a known patterns or *ground truth*. We focus on clustering because it provides an ideal platform for hypothesis generation within the relatively novel area of large-scale medical knowledge mining. Clustering also provides a means to group similar patients and similar segments of patient timelines. This grouping can then provide higher fidelity for supervised tasks like classification or regression since group-wise models that have reduced within-group variance can then be built for each cluster.

However, there are a number of technical challenges that arise in clustering this type and scale of data, notably the multiplicity of possible results and the data size involved. Our work in this thesis addresses some of those challenges as discussed in the following subsections.

1.2.1 Multiplicity of possible results

A potential issue with clustering techniques is the multiplicity of results that can be obtained just by changing the clustering method used or even by varying the parameters of the same clustering method. Since clustering is an unsupervised process and there is no *ground truth*, every clustering outcome is *correct* in some sense. However, not all clusterings are based on underlying patterns; some are more sensitive to certain types of noise.

To address this issue, we focused our efforts on the idea of *consensus clustering*. Consensus clustering methods allow us merge multiple *base clusterings* to form one clustering that, on average, reflects the best agreement with all the base clusterings. These base clusterings may have come from different methods, different clustering parameters or from different feature combinations of the data and consensus clustering attempts to find commonalities between them. In this thesis we generate multiple clusterings by first projecting the data to a random subspace before clustering using the same clustering method: Expectation-Maximization for Gaussian Mixture Models. We focus on using one clustering method so that we can channel our efforts to consensus clustering because the consensus problem for large datasets has not received significant attention unlike the clustering problem which has been heavily addressed over the last five decades.

1.2.2 Data Size

Given the noisiness of data and diversity in patients, a lot of data necessarily needs to be studied to be able to extract statistically significant results. This scale of data is typically larger than the amount of memory installed on commodity computers[3]. Many clustering methods therefore fail since a lot of the methods require the entire dataset to be loaded into memory. Another weakness of existing clustering methods lies in the required multiple passes through the data that result in very long run times for large amounts of data.

There are three ways to solve this problem. First, it may be possible to sample the data and perform analysis on the sample. This approach runs the risk of selecting a non-representative sample and obtaining inaccurate results. Second, the analysis could be run on powerful computers with large installed memory. Unfortunately, such computers are expensive and may be out of the reach of many research groups. A third approach is to develop distributed methods that can run on an large number of commodity machines. We favor this approach and pursue it in this thesis because commodity machines are typically more accessible than single, powerful computers. It is also possible to, with minimal effort, massage such methods to run on volunteer compute nodes as well as run in streaming fashion.

The solutions we present in this thesis are of two classes. First, we implement distributed versions of the popular Expectation-Maximization algorithm for Gaussian Mixture Models and Multinomial Mixture Models (MMM). Our focus with both methods is to deliver timely clustering and consensus clustering results given a certain amount of computing resources. However, both these methods are unable to handle certain data sizes given a fixed amount of computing resources. Second, to deal with data sizes where the MMM with Distributed EM algorithm fails, we have also developed the Streaming Consensus Clustering method that is able to process

[3] e.g. a dual core, 4GB server/workstation at a cost of $1,500. The CSAIL internal cloud currently has over 1,500 virtual cores and 3 TB of memory which we use to construct several "commodity" nodes

any amount of data on any number of computing nodes and deliver timely results with reasonable accuracy.

1.3 Data Ownership

A number of non-technical challenges also affect the ability of researchers to extract knowledge from medical data. These non-technical challenges specifically affect the availability of large datasets for research. While considering the policy challenges, we do not limit ourselves to any specific form of medical data. Our analyses and conclusions here examine these policy challenges from the perspective of all forms of medical data including physiological signals, clinical information such as medication and doctors' notes as well as genomic data. Specifically, in this thesis, we focus on question of medical data ownership.

This thesis reviews the legal stance on medical data ownership in United States common law and state law. It considers how the law has evolved to account for the changing definition of what constitutes a medical record ranging from x-rays to other types of data such as ECG. Also of particular interest, is how the law has evolved to deal with the shift from paper to electronic medical records. Finally, it makes and justifies recommendations for Federal medical data ownership laws as well as patient and healthcare provider co-ownership of property rights to such data.

1.4 Organization

The thesis is organized as follows:

Chapter Two describes our implementation of a distributed clustering algorithm that is able to produce multiple clusterings of the same data set by projecting the dataset into random subspaces. It presents results on clustering a large synthetic dataset. It also highlights performance metrics such as computation time and memory usage.

Chapter Three presents two approaches to consensus clustering. The first approach relies on modelling the consensus clustering problem as a mixture of multinomial distributions and identifies the consensus clusters using a distributed implementation of the Expectation-Maximization algorithm. The second approach is a more general framework that involves progressively sampling the data points with adaptive replacement till all the data points have been clustered. The goal with both algorithms is to produce consensus clusterings that are both timely and accurate. Both approaches are therefore compared theoretically and empirically by presenting results that highlight the trade-offs in terms of accuracy, memory requirements and running time (wall-clock time).

Chapter Four discusses the data ownership considerations that affect the availability of medical data for research. It provides a historical overview of legal approaches to the data ownership question, analyzing the flaws and inconsistencies in these approaches by presenting an overview of the common law approach as well as a survey of the current landscape of US state law. It then proceeds to make recommendations for an approach that fixes these issues by striking an appropriate balance between efficiency and equity.

Chapter 2

Distributed Clustering of Large Datasets

As we mentioned in Chapter 1, the idea of clustering very large datasets is central to this thesis. In the past years, several approaches to both clustering and large-scale clustering have been developed. CURE (Guha et al., 1998) represents clusters as multiple points that have been shrunk towards the center and this approach was used to cluster over 100,000 data points. BIRCH (Zhang et al., 1996), which was also used to cluster 100,000 data points, builds a "*CF-tree*" that contains summaries of densely packed regions of the input space and disposes of sparse regions as outliers. DBSCAN (Ester et al., 1996) discovers clusters by finding points that have "*MinPts*" other points within "*Eps*" distance of themselves and denoting the groups as clusters, using this method to cluster over 12,000 data points. More recently, Chitta et al. (2011) developed an approximation to the kernel k-means method (Dhillon et al., 2004) which avoided the expensive computation of the full kernel by "*restricting the cluster centers to a small subspace spanned by a set of randomly sampled data points*". This method was used to cluster over 1.2 million images.

However, none of these methods hit the scale of data (e.g. > 100 million data points) we are targeting in this thesis so we created a distributed implementation of the

Expectation-Maximization (EM) algorithm (Dempster et al., 1977) for Gaussian Mixture Models. We are not the first to perform distributed EM in this fashion. Gu (2008) states that expectation-maximization on all exponential family distributions may be done this way and Lin et al. (2005) uses this approach to perform privacy-preserving clustering on computing nodes that do not share data. However, neither of these authors tackle the specific problems which we address in this thesis:

Data Size: We demonstrate the distributed EM algorithm on a dataset that is two orders of magnitude higher than what we find in the literature.

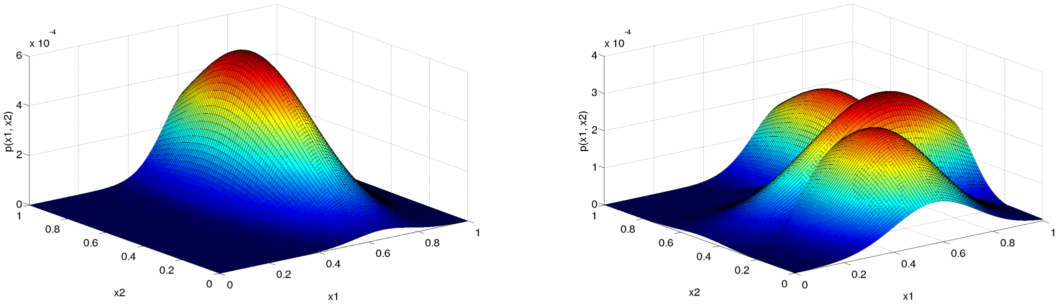
Memory: Due to the sheer size of the data, it is unable to fit entirely in the memory of a single computing node thereby mandating the use of multiple computing nodes.

In the next sections, we describe our implementation of a distributed EM algorithm for Gaussian Mixture Models. Table 2.1 describes the notation used in the subsequent sections.

2.1 Gaussian Mixture Models

Since clustering involves the discovery of groups within data, one approach is to model the data set as points drawn from a mixture of underlying probability distributions. The clustering task then becomes one of discovering the parameters of the underlying distributions. One popular type of distribution for this analysis is the multivariate Gaussian distribution, $\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$. Figure 2-1b shows a mixture of 3 bivariate Gaussian distributions.

The parameters to be learnt are: the means of each Gaussian distribution, $\boldsymbol{\mu}_j$, which reveal where the j component distributions are centered, the covariance matrix of the distributions, $\boldsymbol{\Sigma}_j$, that reveals the dispersion of each model and the ratio of mixing, $\boldsymbol{\pi}_j$, that reveals the likelihood of data points being drawn from each of the component distributions.



(a) Single Bivariate Gaussian Distribution

(b) Mixture of 3 Bivariate Gaussian Distributions

Figure 2-1: Example showing the distribution of a single bivariate Gaussian distribution and a mixture of 3 bivariate distributions

Given an $n \times D$ dataset $\boldsymbol{X}$ drawn from a mixture of m Gaussian distributions, the likelihood of a data point $\boldsymbol{x}_i$ is given by Bishop (2006) as:

$$p(\boldsymbol{x}_i|\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}) = \sum_{j=1}^{m} \pi_j \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu_j}, \boldsymbol{\Sigma_j})$$

where,

$$\mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu_j}, \boldsymbol{\Sigma_j}) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\boldsymbol{\Sigma}_j|^{1/2}} \exp\left(-\frac{1}{2}(\boldsymbol{x}_i - \boldsymbol{\mu}_j)^T \boldsymbol{\Sigma}_j^{-1} (\boldsymbol{x}_i - \boldsymbol{\mu}_j)\right)$$

The likelihood and log-likelihood of the entire dataset is therefore given by:

$$\begin{aligned}
p(\boldsymbol{X}|\boldsymbol{\mu}, \Sigma, \boldsymbol{\pi}) &= \prod_{i=1}^{n} \sum_{j=1}^{m} \pi_j \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu_j}, \boldsymbol{\Sigma_j}) \\
\ln p(\boldsymbol{X}|\boldsymbol{\mu}, \Sigma, \boldsymbol{\pi}) &= \ln \prod_{i=1}^{n} \sum_{j=1}^{m} \pi_j \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu_j}, \boldsymbol{\Sigma_j}) \\
&= \sum_{i=1}^{n} \ln \sum_{j=1}^{m} \pi_j \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu_j}, \boldsymbol{\Sigma_j})
\end{aligned}$$

To determine the optimal parameters, we can attempt to maximize the likelihood by differentiating with respect to $\boldsymbol{\mu}_j$, $\boldsymbol{\Sigma}_j$ and $\boldsymbol{\pi}_j$ and setting the derivatives to zero. This, according to Bishop (2006), leads to the following *maximum likelihood* expressions for

$\boldsymbol{\mu}_j$, $\boldsymbol{\Sigma}_j$ and $\boldsymbol{\pi}_j$ $\forall j \in \{1 \dots m\}$:

$$\begin{aligned}
\boldsymbol{\mu}_j &= \frac{1}{N_j} \sum_{i=1}^{n} \gamma(z_{ij}) \boldsymbol{x_i} \\
\boldsymbol{\Sigma}_j &= \frac{1}{N_j} \sum_{i=1}^{n} \gamma(z_{ij}) (\boldsymbol{x_i} - \boldsymbol{\mu_j})(\boldsymbol{x_i} - \boldsymbol{\mu}_j)^T \\
\boldsymbol{\pi}_j &= \frac{N_j}{n}
\end{aligned}$$

where,

$$\begin{aligned}
N_j &= \sum_{i=1}^{n} \gamma(z_{ij}) \text{ and} \\
\gamma(z_{ij}) &= \frac{\pi_j \mathcal{N}(\boldsymbol{x}_i | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}{\sum\limits_{k=1}^{m} \pi_k \mathcal{N}(\boldsymbol{x}_i | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}
\end{aligned}$$

Note however, that due to the presence of a log of a sum in Eqn. 2.1, the estimate of parameters $\boldsymbol{\mu}_j$, $\boldsymbol{\Sigma}_j$ and π_j contain the term $\gamma(z_{ij})$ which in turn depends on $\boldsymbol{\mu}_j$, $\boldsymbol{\Sigma}_j$ and π_j. This observation however presents an iterative approach to solving for the parameters known as expectation-maximization. The idea is that parameters from the previous iteration are used to calculate $\gamma(z_{ij})$, $\boldsymbol{\mu}_j$, $\boldsymbol{\Sigma}_j$ and $\boldsymbol{\pi}_j$ and the updated parameters are used in the next iteration. This algorithm was shown by Dempster et al. (1977) to increase the likelihood in every iteration and converge. The full EM algorithm is described in Algorithm 1.

We can also introduce the idea of an unobserved $1 \times m$ *latent* variable $\boldsymbol{z}_i$ which determines the particular mixture component a data point $\boldsymbol{x}_i$ was drawn from. In a scenario where $\boldsymbol{x}_i$ is a member of only 1 cluster, $\boldsymbol{z}_i$ contains a 1 in one column and 0 in $m-1$ columns. The marginal distribution of $\boldsymbol{z}_i$ is given by

$$p(z_j = 1) = \pi_j$$

$\gamma(z_{ij})$ can therefore be written as:

$$\gamma(z_{ij}) = \frac{p(z_j = 1)\mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}{\sum\limits_{k=1}^{m} p(z_j = 1)\mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)} = \frac{p(z_j = 1)p(\boldsymbol{x}_i|z_j = 1)}{\sum\limits_{k=1}^{m} p(z_k = 1)p(\boldsymbol{x}_i|z_k = 1)}$$

Written this way, it becomes apparent that $\gamma(z_{ij})$ is the posterior probability of the latent variable. In other words, $\gamma(z_{ij})$ is the probability that the i^{th} point belongs to the j^{th} cluster given the data, and parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$.

```
**Input**: $\boldsymbol{X}$, an $n$ x $D$ matrix of data to be clustered
**Output**: $\boldsymbol{y}$, an $n$ x 1 matrix of cluster labels from $1 \dots m$
Initialize parameters, $\theta^{(0)}$: $\boldsymbol{\mu}^{(0)}$, $\boldsymbol{\Sigma}^{(0)}$, $\boldsymbol{\pi}^{(0)}$
$\mathcal{L}^{(-1)} = -\infty$; $\mathcal{L}^{(0)} = 1$; $a \leftarrow 0$
while $1 - \frac{\mathcal{L}^{(a-1)}}{\mathcal{L}^{(a)}} \geq \epsilon$ do
    // Expectation Step.
    for $i = 1 \to n$ do                                  Can be distributed and performed independently on different nodes
        for $j = 1 \to m$ do
            $p(\boldsymbol{x}_i|\theta^{(a)}, z_{ij} = 1) = \frac{1}{(2\pi)^{\frac{D}{2}}} \frac{1}{|\boldsymbol{\Sigma}_j^{(a)}|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\boldsymbol{x}_i - \boldsymbol{\mu}_j^{(a)})\boldsymbol{\Sigma}_j^{(a),-1}(\boldsymbol{x}_i - \boldsymbol{\mu}_j^{(a)})^T\right)$
        end
    end
    for $i = 1 \to n$ do
        for $j = 1 \to m$ do
            $\gamma(z_{ij}) = p(z_{ij} = 1|\theta^{(a)}, \boldsymbol{x}_i) = \frac{\pi_j^{(a)} p(\boldsymbol{x}_i|\theta^{(a)}, z_{ij})}{\sum_{k=1}^{m} \pi_k^{(a)} p(\boldsymbol{x}_i|\theta^{(a)}, z_{ik})}$
        end

    end
    // Recompute likelihood
    $\mathcal{L}^{(a)} = \sum_{i=1}^{n} \ln \boldsymbol{\pi}^{(a),T} p(\boldsymbol{x}_i|\theta^{(a)}, \boldsymbol{z}_i)$
    // Maximization Step.
    for $j = 1 \to m$ do                                  Partial sums can computed on different nodes, aggregated on central node and re-distributed to nodes
        $\mu_j^{(a+1)} = \frac{1}{\sum_{i=1}^{n} \gamma(z_{ij})} \sum_{i=1}^{n} \gamma(z_{ij}) \boldsymbol{x}_i$
        $\pi_j^{(a+1)} = \frac{1}{n} \sum_{i=1}^{n} \gamma(z_{ij})$
        $\Sigma_j^{(a+1)} = \frac{1}{\sum_{i=1}^{n} \gamma(z_{ij})} \sum_{i=1}^{n} \gamma(z_{ij})(\boldsymbol{x}_i - \boldsymbol{\mu}_j^{(a+1)})^T(\boldsymbol{x}_i - \boldsymbol{\mu}_j^{(a+1)})$
    end
    $a \leftarrow a + 1$
end
for $i = 1 \to n$ do
    $y_i = \arg\max_{j \in 1 \dots m} p(z_{ij} = 1|\theta^{(a)}, \boldsymbol{x}_{\boldsymbol{i}})$
end
//
```

Algorithm 1: EM Algorithm for Gaussian mixture models. See Table 2.1 for a legend of the symbols in the algorithm

Notation	Dimension	Meaning
n	1 x 1	Number of data points
D	1 x 1	Number of input data dimensions
m	1 x 1	Number of desired clusters or distributions in GMM
a	1 x 1	Iteration counter
$\boldsymbol{x}_i$	1 x D	i^{th} row of input data
z_{ij}	1 x 1	Latent variable indicating if the i^{th} row of input data belongs to cluster j, $z_{ij} \in \{0 \ldots 1\}$
$\boldsymbol{\mu}_j$	1 x D	Mean of j^{th} cluster
π_j	1 x 1	Probability of selecting j^{th} cluster
$\boldsymbol{\Sigma}_j$	D x D	Covariance of j^{th} cluster
$\mathcal{L}$	1 x 1	Data likelihood
$\gamma(z_{ij})$	1 x 1	Probability that the i^{th} data point belongs to the j^{th} cluster given the dataset, $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$

Table 2.1: Legend of symbols in Algorithm 1

2.2 Distributed Expectation-Maximization

Since the computation of $p(\boldsymbol{x}_i|\theta^{(a)}, z_{ij})$ and $\gamma(z_{ij})$ as shown in lines 8 and 14 of Algorithm 1 depends on just the parameters of the model (and not the values of other data points), this computation can be parallelized to multiple computing nodes provided each node has the current global parameters ($\boldsymbol{\mu}^{(a)}$, $\boldsymbol{\Sigma}^{(a)}$, $\boldsymbol{\pi}^{(a)}$). Similarly, since the maximization step comprises of summations over the data points, the dataset could be partitioned into non-overlapping sets and summation for each set could happen independently, then the sets are summed themselves. We exploit both opportunities for parallelism in order to implement a shared-nothing, distributed EM framework for GMM that can generate clusterings for an arbitrarily large dataset using a cloud of computers even when the entire dataset cannot entirely reside in the memory of any one computer in the cloud.

The central idea here is to partition the data into non-overlapping sets, transfer these sets to ”slave” nodes on the cloud and perform EM independently on each node. At different points during each iteration, the parameters computed on each node are sent to a “master” central node where they are aggregated and summed before being broadcast back to the compute nodes. The steps involved in developing an implementation of distributed EM are:

Step 0: Decide on the number of instances to use. This depends on the size of the dataset and the amount of memory on every instance. Decide upon the architecture (Section 2.2.1) and protocols e.g `ssh`, `scp` etc. (Section 2.2.4) for communication between nodes.

Step 1: Partition the data (Section 2.2.2)

Step 2: Initialize the parameters (Section 2.2.3) and distribute them to the slaves

Step 3: Start the process on each slave

Step 4: Calculate, on each slave, the conditional probabilities and posteriors for the data points on it. Calculate the partial sum of log-likelihoods and send them to the master.

Step 5: Sum the partial sums of log-likelihoods from the slaves on the master and send the sum to the slaves. If the sum of log-likelihoods satisfy the stopping condition, stop the process on the master and retrieve the cluster label assignments from the slaves.

Step 6: Calculate, on each slave, a partial sum of posterior probabilities and send them to the master. On the master, aggregate these partial sums to form the full sum of posterior probabilities. Then compute the priors on the master and send both the sum of posteriors and the priors to the slaves.

Step 7: On each slave, calculate the partial means and sends them to the master. On the master, aggregate these partial means to form the global means and send them to the slaves.

Step 8: On each slave, calculate partial covariance matrices and sends them to the master. On the master, aggregate these partial covariance matrices to form the global covariance matrices and send them to the slaves. Start the next iteration at Step 4.

Steps 4-8 are detailed in Section 2.2.4. In the subsequent sections, we describe the architecture and niceties of our implementation.

2.2.1 Architecture

In this thesis, we adopt a master-slave topology for distributed learning of the parameters. Therefore, a node is designated as the master node and is responsible for partitioning the data, starting up the slave nodes and aggregating the parameters. The algorithms are implemented in MATLAB[1] because of its superior efficiency in matrix manipulation. Distribution is performed on an internal OpenStack[2] cloud at the Computer Science and Artificial Intelligence Laboratory (CSAIL)[3] at MIT.

2.2.2 Data Partitioning

Our goal here is to create almost equal partitions of the data so each slave would processes an equal-sized partition. Since the data is too large to load into and partition in memory, the data is read in blocks and partitioned on the master node and the partitions are written out to disk. After each block of data is read, it is randomly permuted before roughly equal partitions corresponding to the amount of slave nodes are created. This random permutation is not absolutely necessary since the distributed EM approach we have described so far performs no local approximations of global parameters. Therefore, even if all the data points on a slave belong to a single cluster, the EM results are unaffected. We have, however, chosen to randomize the data partitioning because in our distributed EM implementation, if one or more slaves hold up an aggregation step, the local parameters from the nodes that have completed that step are used to form the global view of the parameters. This implementation detail is explained further in Section 2.2.5. Under this scenario, having some ordering on the data points sent to a slave could affect the results.

[1] http://www.mathworks.com/products/matlab/

[2] http://www.openstack.org/

[3] http://tig.csail.mit.edu/wiki/TIG/OpenStack

2.2.3 Parameter Initialization

According to (Vlassis and Likas, 2002), there is no single accepted way of performing parameter initialization. They recommend that $\boldsymbol{\mu}$ be initialized to random points in the data set and $\boldsymbol{\Sigma}$ set to a spherical covariance with $\sigma^2 = \frac{1}{2D}\min_{i \neq j}||\boldsymbol{\mu}_i - \boldsymbol{\mu}_j||^2$. This strategy would involve loading the entire dataset into memory or at least knowing the size of the dataset a priori. Therefore, for $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ we had to devise a parameter initialization scheme that is compatible with our approach of streaming in the data in blocks.

While streaming the data in, we keep track of the minimum and maximum datapoints in every dimension, $\boldsymbol{x}_{min}$ and $\boldsymbol{x}_{max}$ (both of dimension, $1 \times D$). We then select each $\boldsymbol{\mu}_j$ by randomly sampling the space between the maximum and the minimum in every dimension and we obtain each $\boldsymbol{\Sigma}_j$ by finding the variance of a 2-element dataset that contains only the maximum and the minimum values in every dimension. This variance becomes the diagonal of the initial shared diagonal covariance matrix for every component:

$$\begin{aligned}
\boldsymbol{\mu}_j &= \boldsymbol{x}_{min} + (\boldsymbol{x}_{max} - \boldsymbol{x}_{min}) \text{ diag}(\boldsymbol{rand_{1 \times D}}) \\
\boldsymbol{\Sigma_j} &= \text{diag}\left(\frac{1}{4}(\boldsymbol{x}_{min} - \boldsymbol{x}_{max}) \text{ diag}(\boldsymbol{x}_{min} - \boldsymbol{x}_{max})\right)
\end{aligned}$$

where $\boldsymbol{rand}_{c \times d}$ is a $c \times d$ matrix of random numbers varying from 0 to 1

We initialized the priors on the components as $\pi_k = \frac{1}{m}$ following the approach used by Vlassis and Likas (2002)

2.2.4 Distributed Learning

Communication between the master node and the slaves happens over `ssh`, `scp` and Java[4] sockets. The EM process and other housekeeping on the slave nodes is started using `ssh` and global parameters[5] are transferred to the slaves using master-initiated `scp`. Similarly, local parameters are obtained from the slaves using master-initiated `scp` by making the slaves ping the master when a local view of variables is awaiting collection.

Suppose there are S slaves and a slave, s, receives a subset of the data, $\boldsymbol{X}_s$, such that $|\boldsymbol{X}_s| = n_s$ and each row of $\boldsymbol{X}_s$ is denoted by $\boldsymbol{x}_{s,i}$ the distributed learning approach works as follows:

Step 0 (master): Parameters $\boldsymbol{\mu}^{(0)}$, $\boldsymbol{\Sigma}^{(0)}$ and $\boldsymbol{\pi}^{(0)}$ are initialized at the master and sent to the slaves. Iteration counter, a, is also initialized to 0.

Step 1 (slave): On each slave, s, and for each data point i, and each cluster j, the following is done:

(a) Conditional probabilities are computed:

$$p(\boldsymbol{x}_{s,i}|\theta^{(a)}, z_{ij} = 1) = \frac{1}{(2\pi)^{\frac{D}{2}}} \frac{1}{|\boldsymbol{\Sigma}_j^{(a)}|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\boldsymbol{x}_{s,i} - \boldsymbol{\mu}_j^{(a)})\boldsymbol{\Sigma}_j^{(a),-1}(\boldsymbol{x}_{s,i} - \boldsymbol{\mu}_j^{(a)})^T\right)$$

(b) Posteriors are computed:

$$\gamma_s(z_{ij}) = \frac{\pi_j^{(a)} p(\boldsymbol{x}_{s,i}|\theta^{(a)}, z_{ij})}{\sum\limits_{k=1}^{m} \pi_k^{(a)} p(\boldsymbol{x}_{s,i}|\theta^{(a)}, z_{ik})}$$

[4] http://java.com/en/

[5] 'Parameters' in this section is a looser term and includes $\gamma(z_{ij})$ and $\mathcal{L}$ which are not parameters of the distribution

(c) and the partial sum of log-likelihoods is computed and sent to the master:

$$\mathcal{L}_s^{(a)} = \sum_{i=1}^{n_s} \ln \boldsymbol{\pi}^{(a),T} p(\boldsymbol{x}_{s,i}|\theta^{(a)}, \boldsymbol{z}_i)$$

Step 2 (master): The global sum of log-likelihoods $\mathcal{L}^{(a)}$ is computed. If $1-\frac{\mathcal{L}^{(a-1)}}{\mathcal{L}^{(a)}} < \epsilon$, the computation is stopped and the cluster labels are collected from the slaves. Otherwise, $\mathcal{L}^{(a)}$ is sent to the slaves:

$$\mathcal{L}^{(a)} = \sum_{s=1}^{S} \mathcal{L}_s^{(a)}$$

Step 3 (slave): The partial sum of posteriors, $N_{s,j}$, for each cluster j is computed on each slave, s, and sent to the master:

$$N_{s,j} = \sum_{i=1}^{n_s} \gamma_s(z_{ij})$$

Step 4 (master): The partial sum of posteriors from the slaves is aggregated to form the global sum of posteriors, N_j, for each cluster, j. The new priors for each cluster, $\pi_j^{(a+1)}$ are also recomputed and both posteriors and priors are sent to the slaves:

$$N_j = \sum_{s=1}^{S} N_{s,j}$$

$$\pi_j^{(a+1)} = \frac{N_j}{n}$$

Step 5 (slave): The partial means, $\mu_{s,j}^{(a+1)}$ are calculated and sent to the master:

$$\boldsymbol{\mu}_{s,j}^{(a+1)} = \frac{\sum_{i=1}^{n_s} \gamma_s(z_{ij})\boldsymbol{x}_{s,i}}{N_j} \qquad (2.1)$$

Step 6 (master): The partial means are summed to compute the global mean,

$\mu_j^{(a+1)}$, for each cluster, j, and these are sent to the slaves:

$$\boldsymbol{\mu}_j^{(a+1)} = \sum_{s=1}^{S} \boldsymbol{\mu}_{s,j}^{(a+1)}$$

Note that the denominator in Eqn. 2.1 is the global posterior, N_j, such that summing the partial means at the master is sufficient to compute the global means.

Step 7 (slave): The partial covariance matrices, $\boldsymbol{\Sigma}_{s,j}^{(a+1)}$ are calculated and sent to the master:

$$\boldsymbol{\Sigma}_{s,j}^{(a+1)} = \frac{\sum\limits_{i=1}^{n_s} \gamma_s(z_{ij})(\boldsymbol{x}_{s,i} - \boldsymbol{\mu}_j^{(a+1)})^T(\boldsymbol{x}_{s,i} - \boldsymbol{\mu}_j^{(a+1)})}{N_j}$$

Step 8 (master): The partial covariance matrices are summed to compute the global covariance matrix, $\boldsymbol{\Sigma}_j^{(a+1)}$, for each cluster, j, and these are sent to the slaves:

$$\boldsymbol{\Sigma}_j^{(a+1)} = \sum_{s=1}^{S} \boldsymbol{\Sigma}_{s,j}^{(a+1)}$$

The iteration counter, a, is also incremented i.e. $a \leftarrow a+1$ and the computation loops to Step 1.

2.2.5 Wait time-out and parameter approximation

The master node has to wait for local parameter views at aggregation points; four of them per iteration (log-likelihood (Step 2), posteriors (Step 4), means (Step 6) and covariances (Step 8)) as shown in Figure 2-2. This amounts to four pings and `scp` calls per node per iteration. We must ensure against any one slave's message being lost and the computation left hanging. Therefore, we time-bound the wait at the master nodes. At time-out the global parameter is estimated from the parameters received from S' slaves. Let Φ represent a global parameter to be estimated i.e.

$\Phi \in \{\mathcal{L}^{(a)}, N_j, \boldsymbol{\mu}_j^{(a+1)}, \boldsymbol{\Sigma}_j^{(a+1)}\}$ and $\hat{\Phi}$ be the sum of the local parameters received so far i.e. $\hat{\Phi} = \sum\limits_{s=1}^{S'} \Phi_s$, we estimate Φ as:

$$\Phi = \frac{n\hat{\Phi}}{\sum\limits_{s=1}^{S'} n_s}$$

In our case time-outs were infrequent, happening about once in every run.

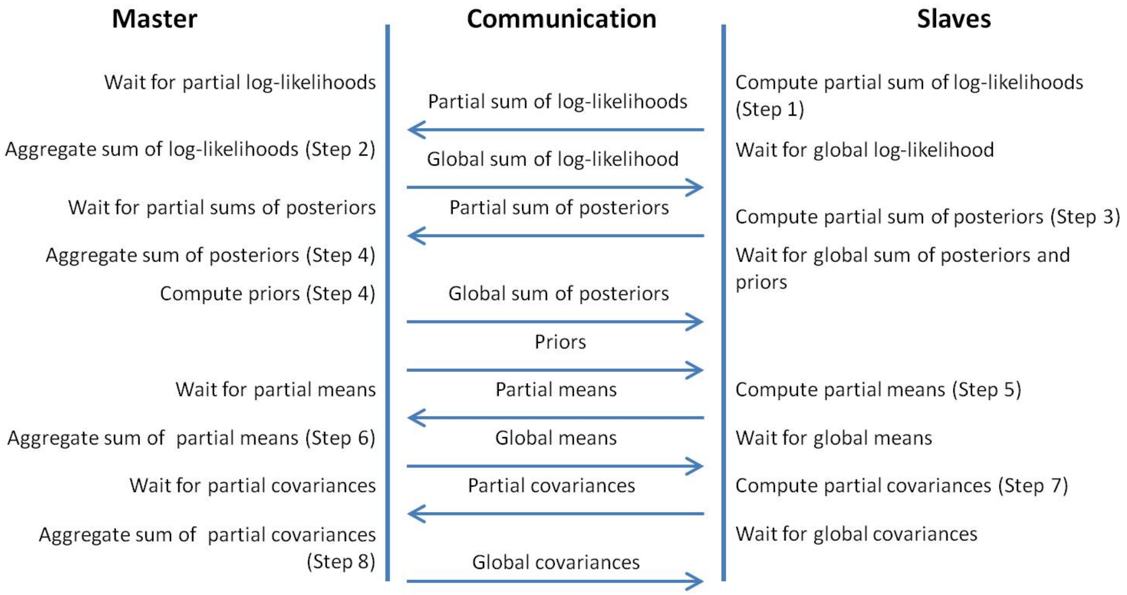

Master
Communication
Slaves
Wait for partial log-likelihoods
Partial sum of log-likelihoods
Compute partial sum of log-likelihoods (Step 1)
Aggregate sum of log-likelihoods (Step 2)
Global sum of log-likelihood
Wait for global log-likelihood
Wait for partial sums of posteriors
Partial sum of posteriors
Compute partial sum of posteriors (Step 3)
Aggregate sum of posteriors (Step 4)
Wait for global sum of posteriors and priors
Compute priors (Step 4)
Global sum of posteriors
Priors
Wait for partial means
Partial means
Compute partial means (Step 5)
Aggregate sum of partial means (Step 6)
Global means
Wait for global means
Wait for partial covariances
Partial covariances
Compute partial covariances (Step 7)
Aggregate sum of partial covariances (Step 8)
Wait for global covariances
Global covariances

Figure 2-2: Master-Slave Communication and Processing during one Distributed GMM Iteration

2.3 Generating Multiple Clusterings

One approach to generating multiple clusterings from the same dataset is to cluster the dataset using a variety of algorithms and varying clustering parameters such as distance measures, feature sets, initializations (Fred and Jain, 2002), and number of clusters (Topchy et al., 2004). To follow this approach would have required scaling up multiple clustering algorithms. In the interest of our time budget, we, instead, explored methods for generating multiple clusterings from our scaled-up expectation-maximization (EM) for Gaussian mixture models (GMM) described in Section 2.2. We generated multiple clusterings by, first, projecting the data to a random subspace, $\mathbf{\Gamma}$, as described in (Topchy et al., 2004), before clustering using our distributed EM for GMM platform. In contrast to (Topchy et al., 2004), we project the data, not just to a 1-dimensional subspace but to a subspace of D' dimensions varying from 1 to D dimensions. The subspace is randomly generated, then Gram-Schmidt orthonormalization is performed to ensure that the basis vectors of the subspace are orthogonal and of unit length. The projected data X_{SS} is thus calculated as

$$\begin{aligned} D' &= \text{ceiling}(1 + {}_{1\times 1}(D-1)) \\ \mathbf{\Gamma} &= \text{qr}(\boldsymbol{rand_{D\times D'}}) \\ \boldsymbol{X_{SS}} &= \boldsymbol{X}\mathbf{\Gamma} \end{aligned}$$

where qr is the Gram-Schmidt orthonormalization operation.

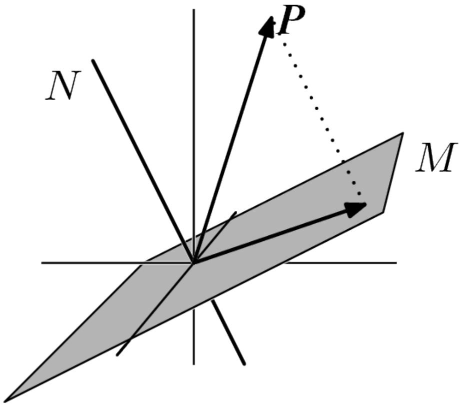


Figure 2-3: Projection of 3-dimensional point P to 2-dimensional plane M[6]

A similar approach was used by Fern and Brodley (2003) before ensemble clustering to demonstrate an alternative to using PCA for dimensionality reduction. A number of authors have also shown that random projection brings out some interesting properties of the data such as making "eccentric" cluster shapes more spherical (Dasgupta, 2000).

[6] Image source: http://commons.wikimedia.org/wiki/File:Linalg_projection_onto_plane_2.png

2.4 Inter-clustering comparison

Throughout this thesis, to evaluate the effectiveness of our algorithms, we need to measure the similarity between two clusterings or between a clustering and the ground truth. To do this, we employ the mutual information between the clusterings as described in (Strehl and Ghosh, 2003). In that paper, the authors utilized the *normalized* mutual information (NMI), using the geometric mean of entropies of both clusterings as the normalization factor. This constrains the NMI to a value between 0 and 1 because the mutual information is observed to be lower than the minimum of the two entropies.

The mutual information between two clusterings, $\boldsymbol{y}_i$ and $\boldsymbol{y}_j$, with cluster labels sampled from $\{1 \ldots F\}$, is given by

$$I(\boldsymbol{y}_i, \boldsymbol{y}_j) = \sum_{f_i=1}^{F} \sum_{f_j=1}^{F} p(f_i, f_j) \log \left(\frac{p(f_i, f_j)}{p(f_i)p(f_j)} \right)$$

and the entropy of a clustering, $\boldsymbol{y}$, is given by

$$H(\boldsymbol{y}) = \sum_{f=1}^{F} p(f) \log p(f)$$

Recalling that the total number of data points is n, if the size of f^{th} cluster in i^{th} clustering is given by n_{f_i} and n_{f_i,f_j} is the number of points that fall both into cluster f_i in clustering $\boldsymbol{y}_i$ and cluster f_j in clustering $\boldsymbol{y}_j$, then the NMI of clusterings $\boldsymbol{y}_i$ and $\boldsymbol{y}_j$ is thus given by

$$
\begin{aligned}
\phi(\boldsymbol{y}_i, \boldsymbol{y}_j) &= \frac{I(\boldsymbol{y}_i, \boldsymbol{y}_j)}{\sqrt{H(\boldsymbol{y}_i)H(\boldsymbol{y}_j)}} \\
&= \frac{\sum\limits_{f_i=1}^{F}\sum\limits_{f_j=1}^{F} p(f_i, f_j)\log\left(\frac{p(f_i,f_j)}{p(f_i)p(f_j)}\right)}{\sqrt{\left(\sum\limits_{f_i=1}^{F} p(f_i)\log p(f_i)\right)\left(\sum\limits_{f_j=1}^{F} p(f_j)\log p(f_j)\right)}} \\
&= \frac{\sum\limits_{f_i=1}^{F}\sum\limits_{f_j=1}^{F} p(f_i|f_j)p(f_j)\log\left(\frac{p(f_i|f_j)p(f_j)}{p(f_i)p(f_j)}\right)}{\sqrt{\left(\sum\limits_{f_i=1}^{F} p(f_i)\log p(f_i)\right)\left(\sum\limits_{f_j=1}^{F} p(f_j)\log p(f_j)\right)}} \\
&= \frac{\sum\limits_{f_i=1}^{F}\sum\limits_{f_j=1}^{F} \frac{n_{f_i,f_j}}{n_{f_j}}\frac{n_{f_j}}{n}\log\left(\frac{\frac{n_{f_i,f_j}}{n_{f_j}}\frac{n_{f_j}}{n}}{\frac{n_{f_i}}{n}\frac{n_{f_j}}{n}}\right)}{\sqrt{\left(\sum\limits_{f_i=1}^{F} \frac{n_{f_i}}{n}\log\frac{n_{f_i}}{n}\right)\left(\sum\limits_{f_j=1}^{F} \frac{n_{f_j}}{n}\log\frac{n_{f_j}}{n}\right)}} \\
&= \frac{\sum\limits_{f_i=1}^{F}\sum\limits_{f_j=1}^{F} n_{f_i,f_j}\log\left(\frac{n_{f_i,f_j}n}{n_{f_i}n_{f_j}}\right)}{\sqrt{\left(\sum\limits_{f_i=1}^{F} n_{f_i}\log\frac{n_{f_i}}{n}\right)\left(\sum\limits_{f_j=1}^{F} n_{f_j}\log\frac{n_{f_j}}{n}\right)}}
\end{aligned}
$$

2.5 Results

2.5.1 Dataset

We employ a synthetic dataset generated using a multivariate Gaussian model $\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ similar to Welling and Kurihara (2006). We set the dimensions $D = 8$ and specify the number of clusters as $m = 10$. We first generate the means for each category ($\boldsymbol{\mu}_j$), then we generate the standard deviations for each dimension such that no two categories are closer than $\tau = \frac{\sigma_i+\sigma_k}{2}$. Thus τ gives us a parameterized way to vary the difficulty level of the task. For our analysis, we set the $\tau = 1$.

2.5.2 Algorithm Verification

We, first of all, verified the equivalence of distributing the EM algorithm by running the non-distributed and distributed versions of the algorithm on the same synthetic dataset of 100,000 points, using the same parameter initializations for both schemes. The distributed algorithm was run on 5 slaves running Ubuntu Linux with 1 core and 2GB of RAM. The master ran on an Ubuntu Linux node with 2 cores and 4GB of RAM. Both schemes came up with the exact same clustering after 51 iterations.

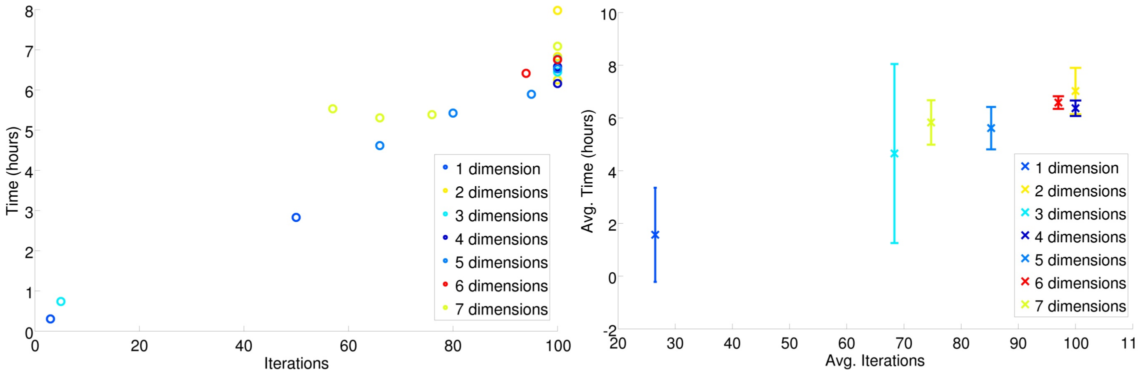
2.5.3 Synthetic Dataset Performance

We then demonstrated the ability of the algorithm to scale up to very large amounts of data and significantly more computing nodes. To do this, we generated 400 million data points using the scheme described in Section 2.5.1. We then partitioned the data 44 ways and ran the distributed EM algorithm on 44 slave nodes running Ubuntu Linux with 4 cores, 8GB of RAM and 90GB of disk space. The master ran on an Ubuntu Linux node with 24 cores, 46GB of RAM and 370GB of disk space. Note that the master could have run on a node with less cores and RAM. All we required was the disk space necessary to store the generated data, its partitions and the results. However, due to inflexibility in sizing of the available nodes, we were constrained to run on such large processing and memory specifications in order to meet our disk space needs.

We generated 20 clusterings for this data, by first projecting the data to a random subspace (with a random number of dimensions < 8) and then clustering using the distributed EM for GMM algorithm. Details of how the data was projected were discussed in Section 2.3.

Computational performance

Figure 2-4 shows the computational performance of each of the 20 clustering tasks, highlighting both the time taken and number of iterations performed and showing how these numbers vary with the number of dimensions in the input dataset. The number of EM iterations performed by each clustering task was limited to 100 and it is apparent from Figure 2-4 that many of the clustering processes were cut off by this limit. A qualitative examination of the figures also indicates that there is more variance in clustering times in low dimensions. This may indicate the presence of more local optima at lower dimensional projections of the input data set.



(a) Clustering times vs. iterations

(b) Average clustering times vs. average iterations

Figure 2-4: EM computational performance in clustering 400 million data points across 44 nodes for 20 different experimental tasks with $m = 10$ clusters

Accuracy

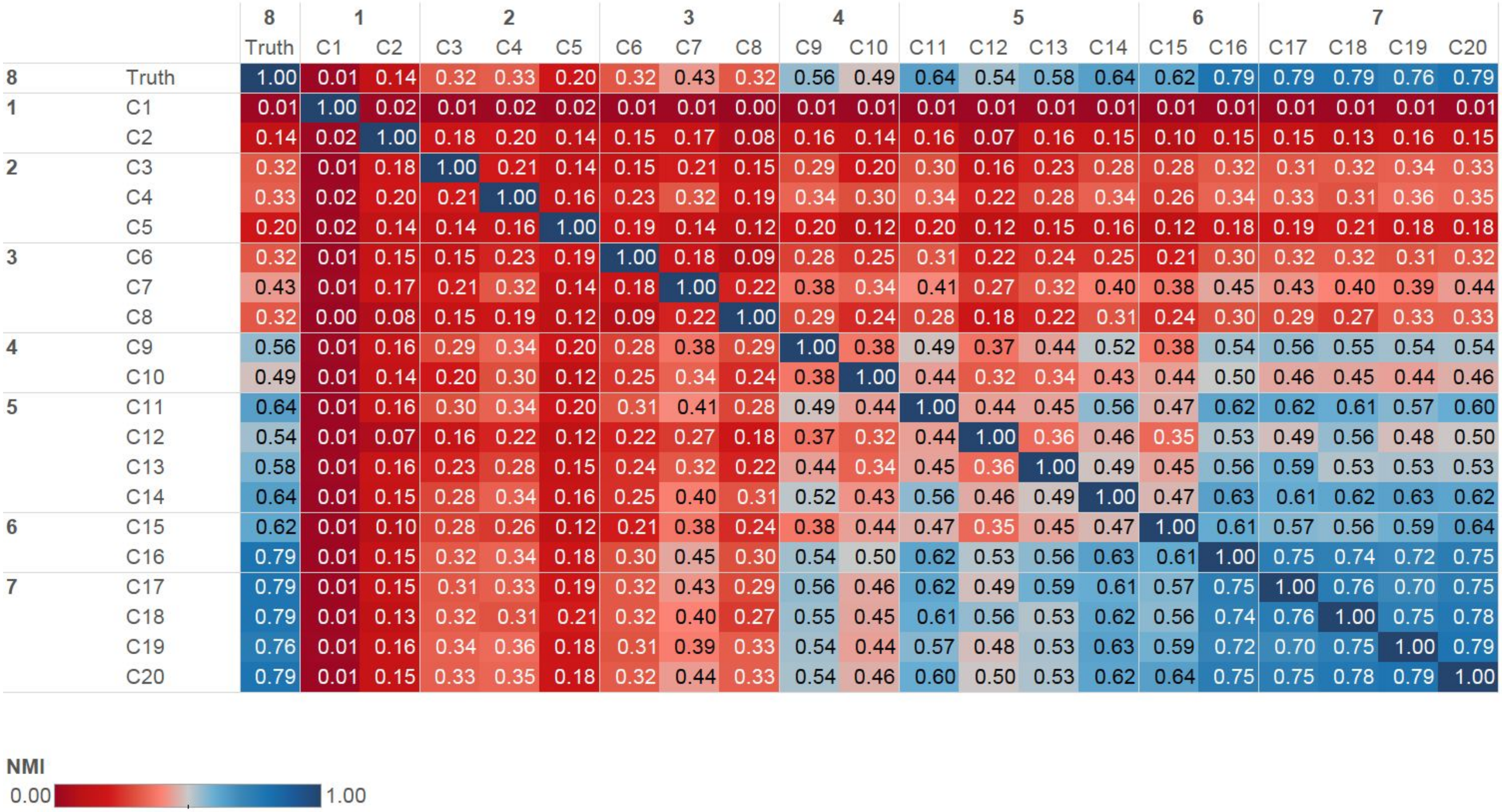
Figure 2-5 shows the pairwise NMI between each of the clusterings as well as the NMI between each clustering and the ground truth. This figure indicates that clusterings that were projected to fewer dimensions agree less with other clusterings and the ground truth.

2.6 Conclusion

We distributed the Expectation-Maximization algorithm for Gaussian Mixture Models and ran it on a very large dataset, demonstrating its performance characteristics. We employed a streaming parameter initialization technique to be able to scale to a dataset that is much larger than the memory of a single commodity computing node. We also utilized a partial update method to enable us overcome any bottlenecks during the communication phase within an EM iteration.

2.7 Future work

One future direction this work could take would be to use other clustering methods in addition to EM for GMM. Some of these methods have been built to handle large datasets e.g. CURE, CLARA, CLARANS, BIRCH and DBSCAN (Xu et al., 2005). Investigation of these methods for the scale of data in this thesis as well as their amenability to distribution on a cloud of computers is therefore worth considering.



		8	1		2			3			4		5				6		7			
		Truth	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	C17	C18	C19	C20
8	Truth	1.00	0.01	0.14	0.32	0.33	0.20	0.32	0.43	0.32	0.56	0.49	0.64	0.54	0.58	0.64	0.62	0.79	0.79	0.79	0.76	0.79
1	C1	0.01	1.00	0.02	0.01	0.02	0.02	0.01	0.01	0.00	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
	C2	0.14	0.02	1.00	0.18	0.20	0.14	0.15	0.17	0.08	0.16	0.14	0.16	0.07	0.16	0.15	0.10	0.15	0.15	0.13	0.16	0.15
2	C3	0.32	0.01	0.18	1.00	0.21	0.14	0.15	0.21	0.15	0.29	0.20	0.30	0.16	0.23	0.28	0.28	0.32	0.31	0.32	0.34	0.33
	C4	0.33	0.02	0.20	0.21	1.00	0.16	0.23	0.32	0.19	0.34	0.30	0.34	0.22	0.28	0.34	0.26	0.34	0.33	0.31	0.36	0.35
	C5	0.20	0.02	0.14	0.14	0.16	1.00	0.19	0.14	0.12	0.20	0.12	0.20	0.12	0.15	0.16	0.12	0.18	0.19	0.21	0.18	0.18
3	C6	0.32	0.01	0.15	0.15	0.23	0.19	1.00	0.18	0.09	0.28	0.25	0.31	0.22	0.24	0.25	0.21	0.30	0.32	0.32	0.31	0.32
	C7	0.43	0.01	0.17	0.21	0.32	0.14	0.18	1.00	0.22	0.38	0.34	0.41	0.27	0.32	0.40	0.38	0.45	0.43	0.40	0.39	0.44
	C8	0.32	0.00	0.08	0.15	0.19	0.12	0.09	0.22	1.00	0.29	0.24	0.28	0.18	0.22	0.31	0.24	0.30	0.29	0.27	0.33	0.33
4	C9	0.56	0.01	0.16	0.29	0.34	0.20	0.28	0.38	0.29	1.00	0.38	0.49	0.37	0.44	0.52	0.38	0.54	0.56	0.55	0.54	0.54
	C10	0.49	0.01	0.14	0.20	0.30	0.12	0.25	0.34	0.24	0.38	1.00	0.44	0.32	0.34	0.43	0.44	0.50	0.46	0.45	0.44	0.46
5	C11	0.64	0.01	0.16	0.30	0.34	0.20	0.31	0.41	0.28	0.49	0.44	1.00	0.44	0.45	0.56	0.47	0.62	0.62	0.61	0.57	0.60
	C12	0.54	0.01	0.07	0.16	0.22	0.12	0.22	0.27	0.18	0.37	0.32	0.44	1.00	0.36	0.46	0.35	0.53	0.49	0.56	0.48	0.50
	C13	0.58	0.01	0.16	0.23	0.28	0.15	0.24	0.32	0.22	0.44	0.34	0.45	0.36	1.00	0.49	0.45	0.56	0.59	0.53	0.53	0.53
	C14	0.64	0.01	0.15	0.28	0.34	0.16	0.25	0.40	0.31	0.52	0.43	0.56	0.46	0.49	1.00	0.47	0.63	0.61	0.62	0.63	0.62
6	C15	0.62	0.01	0.10	0.28	0.26	0.12	0.21	0.38	0.24	0.38	0.44	0.47	0.35	0.45	0.47	1.00	0.61	0.57	0.56	0.59	0.64
	C16	0.79	0.01	0.15	0.32	0.34	0.18	0.30	0.45	0.30	0.54	0.50	0.62	0.53	0.56	0.63	0.61	1.00	0.75	0.74	0.72	0.75
7	C17	0.79	0.01	0.15	0.31	0.33	0.19	0.32	0.43	0.29	0.56	0.46	0.62	0.49	0.59	0.61	0.57	0.75	1.00	0.76	0.70	0.75
	C18	0.79	0.01	0.13	0.32	0.31	0.21	0.32	0.40	0.27	0.55	0.45	0.61	0.56	0.53	0.62	0.56	0.74	0.76	1.00	0.75	0.78
	C19	0.76	0.01	0.16	0.34	0.36	0.18	0.31	0.39	0.33	0.54	0.44	0.57	0.48	0.53	0.63	0.59	0.72	0.70	0.75	1.00	0.79
	C20	0.79	0.01	0.15	0.33	0.35	0.18	0.32	0.44	0.33	0.54	0.46	0.60	0.50	0.53	0.62	0.64	0.75	0.75	0.78	0.79	1.00

Figure 2-5: Pairwise NMI between clusterings generated from projections of the data to random subspaces

Chapter 3

Large-Scale Consensus Clustering

While clustering has its strengths in discovering patterns in unlabelled data, one of its major weaknesses is the multiplicity of possible *correct* results when different clustering algorithms or algorithm parameters are applied to the same dataset (Xu et al., 2005). Unfortunately, since there is typically no ground truth, there is no clear way of ranking the correctness of the clusterings. The approach we employ in this thesis is to create a *consensus clustering* from the multiple diverse base clusterings, the idea being that if there are commonalities shared by the multiple base clusterings, the consensus clustering will highlight them. Specifically, we focus on developing and implementing methods for consensus clustering that are able to handle large amounts of data.

There are a number of methods that may be used to merge multiple clusterings to form a single consensus clustering. Some popular methods include graph partitioning based methods, CSPA, HGPA and MCLA (Strehl and Ghosh, 2003) as well as multidimensional scaling-based methods such as DISTATIS Abdi et al. (2007). In this thesis, we exploit the mixture model method developed in (Topchy et al., 2004) because this method was shown to have more accurate results than graph-based methods HGPA and MCLA and we have discovered, empirically, that it has higher memory efficiency than any of the methods mentioned above.

In this method, multiple clusterings are modelled as a mixture of multinomial models (MMM) and the EM algorithm is used to discover the parameters of the component models and mixing priors. Example input to a consensus clustering algorithm can be found in Table 3.1. We implement a strategy for distributing the EM algorithm developed in (Topchy et al., 2004) to run on a shared-nothing cloud of computers that are able to handle a large amount of data provided sufficient computing resources are available. Wolfe et al. (2008) also uses this approach for multinomial mixture models for use in the natural language processing task of word alignment. The main problem with this approach to distributing the EM algorithm is that for a given amount of computational resources, the algorithm will fail above a particular data size. Also, due to the need to share and maintain global distribution parameters across all nodes, communication bottlenecks could lead to high run times.

To overcome these problems, we additionally developed and implemented an alternative strategy for scaling *any* consensus clustering algorithm including the EM algorithm already mentioned. Unlike the MMM with Distributed EM algorithm, our proposed strategy is able to run on any amount of computational resources from 1 computing node to a cloud of computing nodes and still achieve comparable accuracy and reasonable time performance. We discuss and compare both these approaches in this chapter. In summary our major contributions in this chapter are:

- We develop a novel approach to consensus clustering that resamples the input data and incrementally builds the consensus clustering, demonstrating the ability of this method to perform reasonably without minimum requirements of number of computing nodes or memory. We also test this method by running it on up to 150 cloud computing nodes while processing both a large synthetic dataset of 400 million data points as well as a real-world dataset of over 90,000 images and 10,000 base clusterings of those images.

- We implement a distributed version of the Expectation-Maximization algorithm for consensus clustering, running it on up to 150 cloud computing nodes and processing the same 400 million-point synthetic dataset and real-world dataset of 90,000 images. This implementation allowed us to demonstrate empirically that given a fixed amount of computational resources, this method has an upper-bound on the data sizes it can handle.

3.1 Mixture of Multinomial Models

We model the consensus clustering problem as one of determining the parameters of a mixture of G multinomial models as described in (Topchy et al., 2004). Each multinomial model represents a different consensus cluster in the set of base clusterings, $\boldsymbol{Y}$ where every $1 \times G$ data point in $\boldsymbol{Y}$ corresponds to a $1 \times D$ data point in $\boldsymbol{X}$ from Chapter 2. Table 3.1 shows an example of what the input data to a consensus clustering algorithm looks like. Note that this model requires no correspondence or alignment between cluster labels in different clusterings.

	Base clusterings				
	$g=1$	$g=2$	$g=3$	...	$g=G$
$\boldsymbol{y}_1$	5	1	5	...	3
$\boldsymbol{y}_2$	5	1	3	...	1
$\boldsymbol{y}_3$	5	1	4	...	5
$\boldsymbol{y}_4$	5	1	2	...	2
$\boldsymbol{y}_5$	5	3	5	...	1
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\boldsymbol{y}_n$	2	1	1	...	5

Table 3.1: Example of input to consensus clustering algorithm with G base clusterings and n data points

A data point in the multinomial model is the G-dimensional vector consisting of cluster labels from the G base clusterings for that data point. There are two parameters for describing a mixture of H multinomial models. The parameter $\alpha_h = p(h)$, is the probability of sampling from the h^{th} model and the parameter, $\beta_{hgf} = p_g(f|h)$, is the probability of obtaining a value(label), $f \in \{1 \dots F\}$, for the dimension(clustering), g, given that the data point is from a specific multinomial model, h. The likelihood of a data point, $\boldsymbol{y}_i$ and the log-likelihood of the dataset, $\boldsymbol{Y}$, is given by:

$$p(\boldsymbol{y}_i) = \sum_{j=1}^{H} \alpha_h \prod_{g=1}^{G} \prod_{f=1}^{F} \beta_{hgf}^{\delta(y_{ig},f)}$$

$$p(\boldsymbol{Y}) = \prod_{i=1}^{n} \sum_{j=1}^{H} \alpha_h \prod_{g=1}^{G} \prod_{f=1}^{F} \beta_{hgf}^{\delta(y_{ig},f)}$$

$$\ln p(\boldsymbol{Y}) = \sum_{i=1}^{n} \ln \sum_{j=1}^{H} \alpha_h \prod_{g=1}^{G} \prod_{f=1}^{F} \beta_{hgf}^{\delta(y_{ig},f)}$$

where $\delta(y_{ig}, f) = 1$ if $y_{ig} = f$ and 0 otherwise

3.1.1 Expectation-Maximization

To facilitate performing expectation-maximization for the discovery of the model parameters, latent variables are introduced here. The latent variable indicating membership of a data point in one of H multinomial models is an $n \times H$ matrix where each row contains one column with value 1 and $H-1$ columns with value 0 (assuming each data point must belong to one multinomial model). The update expressions for the parameters $\Theta : \alpha_h, \beta_{hgf}\ \forall h \in \{1 \ldots H\}, g \in \{1 \ldots G\}$ and $f \in \{1 \ldots F\}$ are shown by Topchy et al. (2004) to be:

$$\alpha_h^{(new)} = \frac{\sum\limits_{i=1}^{n} \psi(z_{ih})}{\sum\limits_{k=1}^{H} \sum\limits_{i=1}^{n} \psi(z_{ik})}$$

$$\beta_{hgf}^{(new)} = \frac{\sum\limits_{i=1}^{n} \delta(y_{ig}, f) \psi(z_{ih})}{\sum\limits_{i=1}^{n} \psi(z_{ih})}$$

where $\psi(z_{ih}) = p(z_{ih} = 1 | \Theta^{(old)}, \boldsymbol{y_i}) = \dfrac{\alpha_h^{(old)} \prod\limits_{g=1}^{G} \prod\limits_{f=1}^{F} \left(\beta_{hgf}^{(old)}\right)^{\delta(y_{ig},f)}}{\sum\limits_{k=1}^{H} \alpha_k^{(old)} \prod\limits_{g=1}^{G} \prod\limits_{f=1}^{F} \left(\beta_{kgf}^{(old)}\right)^{\delta(y_{ig},f)}}$

The full EM algorithm is shown in Algorithm 2.

```
Input: $\boldsymbol{Y}$, an $n$ x $G$ matrix of $G$ base clusterings of $n$ data points
Output: $\boldsymbol{r}$, an $n$ x 1 vector of consensus cluster labels from $1 \dots H$
Initialize parameters, $\Theta^{(0)}$: $\boldsymbol{\alpha}^{(0)}$, $\boldsymbol{\beta}^{(0)}$
$\mathcal{L}^{(-1)} = -\infty$; $\mathcal{L}^{(0)} = 1$; $a \leftarrow 0$
while $1 - \frac{\mathcal{L}^{(a-1)}}{\mathcal{L}^{(a)}} \geq \epsilon$ do
    // Expectation Step
    for $i = 1 \to n$ do
        for $h = 1 \to H$ do
            $\psi(z_{ih}) = p(z_{ih} = 1|\Theta^{(a)}, \boldsymbol{y}_i) = \frac{\alpha_h^{(a)} \prod_{g=1}^{G} \prod_{f=1}^{F} \left(\beta_{hgf}^{(a)}\right)^{\delta(y_{ig},f)}}{\sum_{k=1}^{H} \alpha_k^{(a)} \prod_{g=1}^{G} \prod_{f=1}^{F} \left(\beta_{hgf}^{(a)}\right)^{\delta(y_{ig},f)}}$
        end
    end
    // Recompute likelihood
    $\mathcal{L}^{(a)} = \sum_{i=1}^{n} \ln \sum_{h=1}^{H} \alpha_h^{(a)} \prod_{g=1}^{G} \prod_{f=1}^{F} \left(\beta_{hgf}^{(a)}\right)^{\delta(y_{ig},f)}$
    // Maximization Step
    for $h = 1 \to H$ do
        $\alpha_h^{(a+1)} = \frac{\sum_{i=1}^{n} \psi(z_{ih})}{\sum_{k=1}^{H} \sum_{i=1}^{n} \psi(z_{ik})}$
        for $g = 1 \to G$ do
            for $f = 1 \to F$ do
                $\beta_{hgf}^{(a+1)} = \frac{\sum_{i=1}^{n} \delta(y_{ig}, f)\psi(z_{ih})}{\sum_{i=1}^{n} \psi(z_{ih})}$
            end
        end
    end
    $a \leftarrow a + 1$
end
for $i = 1 \to n$ do
    $r_i = \arg\max_{h \in 1 \dots H} \psi(z_{ih})$
end
// $\delta(y_{ig}, f) = 1$ if $y_{ig} = f$ and $0$ otherwise
// See Table 3.2 for a legend of the symbols in the algorithm
```

Algorithm 2: EM Algorithm for multinomial mixture models

Notation	Dimension	Meaning
g	1 x 1	Clustering index
f	1 x 1	Cluster label
F	1 x 1	Maximum cluster label in any base clustering
h	1 x 1	Multinomial model in mixture of multinomial models
H	1 x 1	Number of multinomial models in mixture of multinomial models
$\boldsymbol{y_i}$	1 x G	Cluster assignment of data point i in each of G base clusterings
z_{ih}	1 x 1	Latent variable indicating if the i^{th} row of input data belongs to consensus cluster h
β_{hgf}	1 x 1	Probability of emitting label f from base clustering g if data point belongs to consensus cluster h
α_h	1 x 1	Probability of a data point belonging to the h^{th} consensus cluster

Table 3.2: Legend of symbols in Algorithm 2

3.2 Consensus Clustering for Big Data

We implement two approaches to do consensus clustering on large datasets:

1. Distributing the EM algorithm and performing it synchronously over a cloud of computing nodes
2. Building consensus incrementally by iteratively resampling the input data

3.2.1 MMM with Distributed EM

In similar fashion to the approach described in Section 2.2, the EM algorithm for the multinomial mixture model can be distributed to multiple computing nodes by partitioning the input data into non-overlapping sets and performing the expectation step and part of the maximization step on slave nodes with aggregation of $\mathcal{L}$ and Θ updates at the master. Here again this is enabled by the between-point independence of computing $p(z_{ih} = 1|\Theta^{(a)}, \boldsymbol{y}_i)$ as well as being able to compute partial sums for $\alpha_h^{(a+1)}$ and $\beta_{hgf}^{(a+1)}$ on each slave node and summing these partial sums on the master node. Suppose there are S slaves and a slave, s, receives a subset of the data, $\boldsymbol{Y}_s$, such that $|\boldsymbol{Y}_s| = n_s$ and each row of $\boldsymbol{Y}_s$ is denoted by $\boldsymbol{y}_{s,i}$, the distribution approach can be outlined as follows:

Step 0 (master and slaves): Iteration counter, a, is initialized to 0.

Step 1 (slave): On each slave, s, and for each data point i, and each cluster j, randomly initialize and normalize the posterior probabilities:

$$\boldsymbol{\Omega}_s = \boldsymbol{rand}_{n_s,H}$$

$$\psi_s(z_{ih}) = \frac{\Omega_{s,ih}}{\sum\limits_{k=1}^{H} \Omega_{s,ik}}$$

Recall that $\boldsymbol{rand}_{c\times d}$ is a $c \times d$ matrix of random numbers varying from 0 to 1

Step 2 (slave): The partial sum of posteriors, $\Psi_{s,h}$ for each cluster h is computed on each slave and sent to the master

$$\Psi_{s,h} = \sum_{i=1}^{n_s} \psi_s(z_{ih})$$

Step 3 (master): The partial sum of posteriors from the slaves is aggregated to form the global sum of posteriors, Ψ_h, for each cluster, h and sent to the slaves:

$$\Psi_h = \sum_{s=1}^{S} \Psi_{s,h}$$

Step 4 (slave): The priors, α_h, are calculated:

$$\alpha_h^{(a+1)} = \frac{\Psi_h}{\sum\limits_{k=1}^{H} \Psi_k}$$

Step 5 (slave): The local views of parameter, $\beta_{s,hgf}^{(a+1)}$, are calculated and sent to the master:

$$\beta^{(a+1)}_{s,hgf} = \frac{\sum\limits_{i=1}^{n_s} \delta(y_{s,ig}, f)\psi_s(z_{ih})}{\boldsymbol{\psi}_h} \tag{3.1}$$

Step 6 (master): The partial $\beta^{(a+1)}_{s,hgf}$'s are summed to compute the global $\beta^{(a+1)}_{hgf}$ and this is sent to the slaves:

$$\beta^{(a+1)}_{hgf} = \sum_{s=1}^{S} \beta^{(a+1)}_{s,hgf}$$

Step 7 (slave) The slaves are then able to update their posteriors for data points $i \in \{1 \dots n_s\}$:

$$\psi_s(z_{ih}) = \frac{\alpha^{(a+1)}_h \prod\limits_{g=1}^{G} \prod\limits_{f=1}^{F} \left(\beta^{(a+1)}_{hgf}\right)^{\delta(y_{s,ig}, f)}}{\sum\limits_{k=1}^{H} \alpha^{(a+1)}_k \prod\limits_{g=1}^{G} \prod\limits_{f=1}^{F} \left(\beta^{(a)}_{hgf}\right)^{\delta(y_{s,ig}, f)}}$$

and the partial sum of log-likelihoods is computed and sent to the master:

$$\mathcal{L}^{(a+1)}_s = \sum_{i=1}^{n_s} \ln \sum_{h=1}^{H} \alpha^{(a+1)}_h \prod_{g=1}^{G} \prod_{f=1}^{F} \left(\beta^{(a+1)}_{hgf}\right)^{\delta(y_{ig}, f)}$$

Step 8 (master): The global sum of log-likelihoods $\mathcal{L}^{(a+1)}$ is computed. If $1 - \frac{\mathcal{L}^{(a)}}{\mathcal{L}^{(a+1)}} < \epsilon$, the computation is stopped and the cluster labels are collected from the slaves. Otherwise, $\mathcal{L}^{(a+1)}$ is sent to the slaves:

$$\mathcal{L}^{(a+1)} = \sum_{s=1}^{S} \mathcal{L}^{(a+1)}_s$$

The iteration counter, a, is also incremented i.e. $a \leftarrow a + 1$ and the computation loops to Step 1.

3.2.2 Building Consensus Clustering via Resampling

One of the main contributions of this thesis is Streaming Consensus Clustering, an intuitive approach to performing large-scale consensus clustering given limited com-

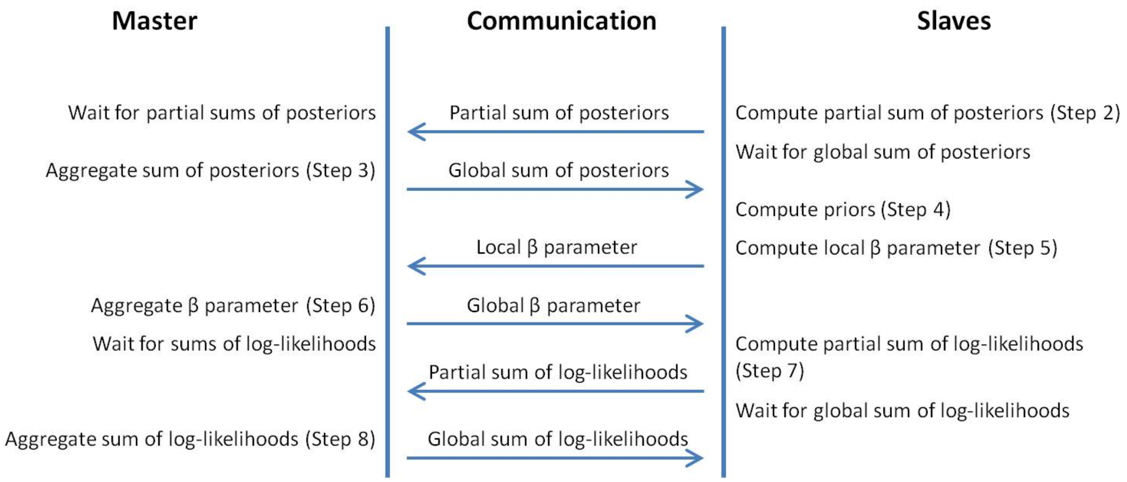

Master
Communication
Slaves
Wait for partial sums of posteriors
Partial sum of posteriors
Compute partial sum of posteriors (Step 2)
Wait for global sum of posteriors
Aggregate sum of posteriors (Step 3)
Global sum of posteriors
Compute priors (Step 4)
Local β parameter
Compute local β parameter (Step 5)
Aggregate β parameter (Step 6)
Global β parameter
Wait for sums of log-likelihoods
Compute partial sum of log-likelihoods (Step 7)
Partial sum of log-likelihoods
Wait for global sum of log-likelihoods
Aggregate sum of log-likelihoods (Step 8)
Global sum of log-likelihoods

Figure 3-1: Master-Slave Communication and Processing during one MMM with Distributed EM Iteration

putational resources. While the previous approach of distributing the EM algorithm outlined in Section 3.2.1 can be limited by number of slaves available and the amount of memory on each slave, this proposed method has the following advantages:

- It would work on any number of nodes including a single commodity node.
- This approach is also useful for performing consensus clustering when the input base clusterings are streamed in.
- This approach allows us plug in other consensus clustering methods that would have otherwise been limited by available computational resources. In this thesis, we demonstrate its usage with the Expectation-Maximization method for consensus clustering (Topchy et al., 2004).

Notation for Streaming Consensus Clustering

In addition to the notation already described, we define 3 subsets of the base clusterings, $\boldsymbol{Y}$ (Table 3.1 describes what this input looks like):

- The *resolved set*, $\boldsymbol{U}$
- The *current set*, $\boldsymbol{W}$, and

- The *anchor set* $\boldsymbol{V}$

	Clusterings			
Idx	**1**	**2**	**3**	**4**
1	2	1	3	2
5	1	3	3	1
7	2	2	3	3
10	2	1	2	3
15	1	1	2	3
16	1	3	2	2
17	2	3	1	3
20	1	2	2	1
22	1	3	3	3
25	3	3	2	2
26	3	1	3	1
28	3	3	1	1
29	2	2	1	3
31	2	2	2	1
32	1	3	3	2

(a) Remaining dataset, $\boldsymbol{Y}^{(a)}$

	Clusterings				
Idx	**1**	**2**	**3**	**4**	**Lbl**
2	1	1	1	2	2
3	1	2	2	3	2
4	3	3	3	1	1
6	1	1	3	2	3
8	2	2	3	1	3
9	3	1	2	3	3
12	1	2	2	2	1
13	3	1	3	2	1
23	2	1	1	1	2
24	1	3	1	1	3
27	3	3	3	1	3
30	3	3	2	2	1
33	1	3	2	1	2
34	3	2	3	2	1
35	2	2	1	3	2

(b) Resolved set, $\boldsymbol{U}^{(a)}$

	Clusterings				
Idx	**1**	**2**	**3**	**4**	**Lbl**
4	3	3	3	1	1
6	1	1	3	2	3
9	3	1	2	3	3
13	3	1	3	2	1
23	2	1	1	1	2
33	1	3	2	1	2

(c) Anchor set, $\boldsymbol{V}$

	Clusterings			
Idx	**1**	**2**	**3**	**4**
11	2	3	1	1
14	1	1	3	2
18	1	2	2	3
19	2	2	3	2
21	2	1	2	2

(d) Current set, $\boldsymbol{W}$

Table 3.3: Sample dataset with 35 data points, 4 base clusterings and 3 consensus clusters showing the remaining dataset after sampling as well as the resolved set(highlighting anchor indices), anchor set and current set

Table 3.3 shows a sample dataset that has been partitioned into sets $\boldsymbol{U}^{(a)}$, $\boldsymbol{V}$ and $\boldsymbol{W}$ after a iteration steps. As is apparent from the table, $\boldsymbol{V} \subset \boldsymbol{U}^{(a)}$. Also note that the indexes used in each of these sets are indices in $\boldsymbol{Y}$ i.e. global, thus they have no relation to the size of the set. For instance, $\boldsymbol{U}^{(a)}$ contains index 35 even though it contains only 15 points.

Recalling that the number of base clusterings is given by G, the dimensions of these sets are $|\boldsymbol{U}| \times G$, $|\boldsymbol{W}| \times G$ and $|\boldsymbol{V}| \times G$, respectively. Also, recalling that the desired number of consensus clusters is given by H, let $\boldsymbol{c}_{\boldsymbol{U},j}$ and $\boldsymbol{c}_{\boldsymbol{V},j}$ represent the indices (in $\boldsymbol{Y}$) of the members of $\boldsymbol{U}^{(a)}$ and $\boldsymbol{V}$ respectively, that belong to the j^{th} consensus cluster. Similarly, let $\boldsymbol{c}_{\boldsymbol{W},j}$ and $\boldsymbol{c}_{\boldsymbol{V}\cup\boldsymbol{W},j}$ represent the indices of the data points that belong to the j^{th} cluster of $\boldsymbol{W}$ and $\boldsymbol{V} \cup \boldsymbol{W}$ respectively. For instance, in Table 3.3, $\boldsymbol{c}_{\boldsymbol{U},1} = \{4, 12, 13, 30, 34\}$.

Algorithm

Initialization Step: Randomly sample the rows of $\boldsymbol{Y}$ without replacement to initialize the subset, $\boldsymbol{U}^{(0)}$. Then, perform consensus clustering on this subset to generate $\boldsymbol{c}_{\boldsymbol{U},j}\ \forall j \in \{1 \dots H\}$ using an algorithm like Expectation-Maximization as described in Section 3.1. If the data is ordered in some way, a full read pass over the data may be required to ensure that the entire input space is properly sampled[1]. The size of this sample, $|\boldsymbol{U}^{(0)}|$, should be such that the sample can be clustered in-memory using the chosen method.

Since this sampling is without replacement, set $\boldsymbol{Y}^{(0)} \leftarrow \boldsymbol{Y} - \boldsymbol{U}^{(0)}$. Also, initialize iteration counter: $a \leftarrow 0$. Table 3.4 shows an example of initial input data and Table 3.5 shows the initial sample and remaining input data.

Step 1: Sample from input data Sample a random subset $\boldsymbol{W}$ from the remaining input data, $\boldsymbol{Y}^{(a)}$. Let $\boldsymbol{b}$ represent the indices in $\boldsymbol{Y}$ of the data points in $\boldsymbol{W}$. Initialize a $|\boldsymbol{W}| \times H$ votes matrix $\boldsymbol{T}$ to all zeros. The columns of $\boldsymbol{T}$ represent the number of votes for each consensus cluster and the rows represent the data points in the current set. Set $\boldsymbol{Y}^{(a+1)} \leftarrow \boldsymbol{Y}^{(a)} - \boldsymbol{W}$. Table 3.6 shows the current set sample and the remaining input data from the data in Table 3.5.

Step 2: Sample from resolved set : Sample *p representatives* from each $\boldsymbol{c}_{\boldsymbol{U},j}$ which, recall, are the indices (in $\boldsymbol{Y}$) of the members of $\boldsymbol{U}^{(a)}$ that belong to the j^{th} consensus cluster. These *representatives*, $\boldsymbol{c}_{\boldsymbol{V},j}\ \forall j \in \{1 \dots H\}$ form sets of indices that are used to select a new anchor set $\boldsymbol{V}$ from $\boldsymbol{U}^{(a)}$. Tables 3.7 and 3.9 show the set of representatives for 2 resamples of the resolved set in Table 3.5.

Step 3: Consensus : Perform consensus clustering on $\boldsymbol{V} \cup \boldsymbol{W}$ and build the set of indices for each cluster found: $\boldsymbol{c}_{\boldsymbol{V}\cup\boldsymbol{W},j}\ \forall j \in \{1 \dots H\}$.

Step 4: Vote : Update the votes array by incrementing the votes for each data point in the column representing the anchor set cluster that was most represented in

[1] If the data is ordered but not fully read, it is possible that the initial resolved set would not contain any samples from one or more clusters

the current set cluster where the data point was found as follows:

(a) Count the number of members, cc_{jk}, from each anchor set cluster, k, that falls in each resulting cluster, j, found in $\boldsymbol{V} \cup \boldsymbol{W}$ as per step 3:

$$cc_{jk} = |\boldsymbol{c}_{\boldsymbol{V}\cup\boldsymbol{W},j} \cap \boldsymbol{c}_{\boldsymbol{V},k} \ \forall j,k \in 1\dots H|$$

(b) For each data point, $\boldsymbol{w}_i \in \boldsymbol{W}$, get its corresponding index in $\boldsymbol{Y}$, $\boldsymbol{b}_i$

(c) Locate the cluster, l, where $\boldsymbol{w}_i$ falls: $l = \underset{k\in\{1\dots H\}}{\arg\max} |\, \boldsymbol{c}_{\boldsymbol{V}\cup\boldsymbol{W},k} \cap \boldsymbol{b}_i|$

(d) Update the appropriate cell in $\boldsymbol{T}$:

$$\boldsymbol{T}_{ih} = \boldsymbol{T}_{ih} + 1$$

where $h = \underset{k\in\{1\dots H\}}{\arg\max} |\, \boldsymbol{c}_{\boldsymbol{V}\cup\boldsymbol{W},l} \cap \boldsymbol{c}_{\boldsymbol{V},k}|$

Tables 3.8 and 3.10 show the intersection count, cc_{jk}, and votes array, $\boldsymbol{T}$, for clustering the current set in Table 3.6 and the anchor sets in Tables 3.7 and 3.9.

Step 5: Rinse, repeat : Perform steps 2 - 5, B times in total.

Step 6: Merge : Merge the current set and resolved set i.e. $\boldsymbol{U^{a+1}} \leftarrow \boldsymbol{U^{(a)}} \cup \boldsymbol{W}$. Also merge the cluster indexes of the resolved set and current set by assigning the data points in the current set to the clusters in the resolved set that received the most votes in $\boldsymbol{T}$ i.e. $\boldsymbol{c_{U,j}} \leftarrow \boldsymbol{c_{U,j}} \cup i$ if $j = \underset{k\in 1\dots H}{\arg\max}\, T_{i,k}$. Table 3.11 shows the new resolved set for the process described in Tables 3.6 - 3.10.

Step 7: Stop? : If $|\boldsymbol{Y}| = 0$ then stop or else go to Step 1.

Idx	**Clusterings**			
1	2	1	1	2
2	1	2	2	2
3	2	1	2	1
4	1	2	2	1
5	2	2	1	2
6	1	2	1	1
7	2	1	1	2
8	1	1	2	1
9	1	1	1	2
10	2	1	1	2
11	2	1	1	2
12	1	2	1	1
13	1	1	2	2
14	2	1	1	1
15	1	1	2	2
16	1	1	1	1
17	2	1	1	2
18	1	1	2	1
19	1	1	2	2
20	1	2	2	2
21	2	1	2	1
22	1	1	2	2
23	2	2	2	1
24	1	1	1	2
25	1	1	1	1
26	1	2	2	1
27	1	2	2	1
28	2	1	2	1
29	2	1	1	2
30	2	2	2	2
31	2	2	2	2
32	2	1	1	1
33	2	2	2	2
34	1	1	2	1
35	2	2	1	1

Table 3.4: $n = 35$ dataset input to consensus clustering algorithm. Number of base clusterings, $G = 4$, maximum cluster label, $F = 2$

Having presented a basic framework for streaming through the data and building a consensus clustering, we now address choices we have to make for the parameters and a simple extension to form our final algorithm.

Idx	Clusterings				Lbl
3	2	1	2	1	1
4	1	2	2	1	2
5	2	2	1	2	2
6	1	2	1	1	2
12	1	2	1	1	2
18	1	1	2	1	1
20	1	2	2	2	2
21	2	1	2	1	1
29	2	1	1	2	2
35	2	2	1	1	2

(a) Resolved set, $\boldsymbol{U}^{(0)}$

Idx	Clusterings			
1	2	1	1	2
2	1	2	2	2
7	2	1	1	2
8	1	1	2	1
9	1	1	1	2
10	2	1	1	2
11	2	1	1	2
13	1	1	2	2
14	2	1	1	1
15	1	1	2	2
16	1	1	1	1
17	2	1	1	2
19	1	1	2	2
22	1	1	2	2
23	2	2	2	1
24	1	1	1	2
25	1	1	1	1
26	1	2	2	1
27	1	2	2	1
28	2	1	2	1
30	2	2	2	2
31	2	2	2	2
32	2	1	1	1
33	2	2	2	2
34	1	1	2	1

(b) Remaining input data, $\boldsymbol{Y}^{(0)}$

Table 3.5: Streaming Consensus Clustering Initialization Step showing $\boldsymbol{U}^{(0)}$ and $\boldsymbol{Y}^{(0)}$. As is evident, $\boldsymbol{c}_{\boldsymbol{U},1} = \{3, 18, 21\}$ and $\boldsymbol{c}_{\boldsymbol{U},2} = \{4, 5, 6, 12, 20, 29, 35\}$

Choosing Size of Anchor Set

The size of the anchor set is determined by the number of *representatives* sampled from each of the resolved set clusters. We set the number of representatives per cluster as a fraction, η, of the smallest cluster size i.e. $p = \eta \min(|\boldsymbol{c}_{\boldsymbol{U},j}|)$. This allows us sample equally from all clusters in the resolved set. As the resolved set grows, sampling a fraction of even the smallest cluster will become impractical so the maximum p possible is limited to a constant. We set this constant as a fraction of the current set size so that the mixture ratio between the anchor set and the current

Idx	**Clusterings**			
7	2	1	1	2
13	1	1	2	2
14	2	1	1	1
17	2	1	1	2
19	1	1	2	2
28	2	1	2	1

(a) Current set, $\boldsymbol{W}$

Idx	**Clusterings**			
1	2	1	1	2
2	1	2	2	2
8	1	1	2	1
9	1	1	1	2
10	2	1	1	2
11	2	1	1	2
15	1	1	2	2
16	1	1	1	1
22	1	1	2	2
23	2	2	2	1
24	1	1	1	2
25	1	1	1	1
26	1	2	2	1
27	1	2	2	1
30	2	2	2	2
31	2	2	2	2
32	2	1	1	1
33	2	2	2	2
34	1	1	2	1

(b) Remaining input data, $\boldsymbol{Y}^{(1)}$

Table 3.6: Streaming Consensus Clustering Step 1 showing $\boldsymbol{W}$ and $\boldsymbol{Y}^{(1)}$. $\boldsymbol{T}$ is initialized to a 6×2 matrix of zeros

Idx	**Clusterings**				**Lbl**
3	2	1	2	1	1
5	2	2	1	2	2
6	1	2	1	1	2
18	1	1	2	1	1

Table 3.7: Streaming Consensus Clustering sample from resolved set, for $b = 1$ showing anchor set $\boldsymbol{V}$ obtained from sets of indices, $\boldsymbol{c}_{\boldsymbol{V},1} = \{3, 18\}$ and $\boldsymbol{c}_{\boldsymbol{V},2} = \{5, 6\}$

set can be controlled.

Streaming Consensus Clustering

The amenability of the algorithm described in Section 3.2.2 to streaming is not immediately apparent since the entire input data, $\boldsymbol{Y}$, and resolved set, $\boldsymbol{U}$ seem to reside entirely in memory. Therefore, to utilize the streaming properties of the algorithm,

		k	
		1	**2**
j	**1**	2	0
	2	0	2

(a) Intersection count, cc_{jk}

1	**2**
0	1
1	0
0	1
0	1
1	0
1	0

(b) Votes array, $\boldsymbol{T}$

Table 3.8: Streaming Consensus Clustering Vote Step, for $b = 1$ showing cc_{jk} and votes array $\boldsymbol{T}$ obtained from clustering results, $\boldsymbol{c}_{\boldsymbol{V}\cup\boldsymbol{W},1} = \{3, 13, 18, 19, 28\}$ and $\boldsymbol{c}_{\boldsymbol{V}\cup\boldsymbol{W},2} = \{5, 6, 7, 14, 17\}$

Idx	**Clusterings**				**Lbl**
3	2	1	2	1	1
20	1	2	2	2	2
21	2	1	2	1	1
29	2	1	1	2	2

Table 3.9: Streaming Consensus Clustering sample from resolved set, for $b = 2$ showing anchor set $\boldsymbol{V}$ obtained from sets of indices, $\boldsymbol{c}_{\boldsymbol{V},1} = \{3, 21\}$ and $\boldsymbol{c}_{\boldsymbol{V},2} = \{20, 29\}$

		k	
		1	**2**
j	**1**	2	1
	2	0	1

(a) Intersection count, cc_{jk}

1	**2**
1	1
1	1
1	1
1	1
1	1
2	0

(b) Votes array, $\boldsymbol{T}$

Table 3.10: Streaming Consensus Clustering Vote Step, for $b = 2$ showing cc_{jk} and votes array $\boldsymbol{T}$ obtained from clustering results, $\boldsymbol{c}_{\boldsymbol{V}\cup\boldsymbol{W},1} = \{3, 7, 14, 17, 21, 28, 29\}$ and $\boldsymbol{c}_{\boldsymbol{V}\cup\boldsymbol{W},2} = \{13, 19, 20\}$

the following modifications must be made.

- The initial resolved set, $\boldsymbol{U}^{(0)}$, and subsequent current sets, $\boldsymbol{W}$, are read (rather than sampled) from the stream (or disk). The amount of data read from the stream is determined by how much data can be clustered in-memory by the chosen method.

- After step 7 in 3.2.2 where a new resolved set is generated by merging with the

Idx	Clusterings				Lbl
3	2	1	2	1	1
4	1	2	2	1	2
5	2	2	1	2	2
6	1	2	1	1	2
7	2	1	1	2	1
12	1	2	1	1	2
13	1	1	2	2	1
14	2	1	1	1	1
17	2	1	1	2	1
18	1	1	2	1	1
19	1	1	2	2	1
20	1	2	2	2	2
21	2	1	2	1	1
28	2	1	2	1	1
29	2	1	1	2	2
35	2	2	1	1	2

Table 3.11: Streaming Consensus Clustering Merge Step showing $\boldsymbol{U}^{(0)}$. As is evident, $\boldsymbol{c}_{\boldsymbol{U},1} = \{3, 7, 13, 14, 17, 18, 19, 21, 28\}$ and $\boldsymbol{c}_{\boldsymbol{U},2} = \{4, 5, 6, 12, 20, 29, 35\}$

current set ($\boldsymbol{U}^{a+1} \leftarrow \boldsymbol{U}^{(a)} \cup \boldsymbol{W}$), the resolved set is sampled without replacement and the sample written to the output stream (or disk) i.e

$$\text{Sample } \boldsymbol{out} \sim \boldsymbol{U}^{a+1} \text{ where, } |\boldsymbol{out}| = |\boldsymbol{W}|$$
$$\boldsymbol{U}^{a+1} \leftarrow \boldsymbol{U}^{a+1} - \boldsymbol{out}$$

For this procedure to function properly, we assume that there is no ordering on the clusters in the input stream. We also recognize that this modified form of sampling is not probabilistically equivalent to sampling over the entire dataset. This approach biases the sampling towards more recent input values. This property may actually be useful for applications where there is a temporal relationship between values in the input stream and deserves further investigation.

```
Input: $\boldsymbol{Y}^{(0)}$, an $n$ x $G$ matrix of $G$ base clusterings of $n$ data points
Output: $\boldsymbol{r}$, an $n$ x 1 matrix of consensus cluster labels from $1 \ldots H$
Sample $\boldsymbol{U}^{(0)} \sim \boldsymbol{Y}^{(0)}$
$\boldsymbol{b} \leftarrow$ indices in $\boldsymbol{Y}$ of $\boldsymbol{W}$
$\boldsymbol{Y}^{(1)} \leftarrow \boldsymbol{Y}^{(0)} - \boldsymbol{U}^{(0)}$
$\{\boldsymbol{c}_{U,1} \ldots \boldsymbol{c}_{U,H}\} \leftarrow \text{Cluster}(\boldsymbol{U}^{(0)})$
$a \leftarrow 0$
while $|\boldsymbol{Y}^{(a)}| > 0$ do
    Sample $\boldsymbol{W} \sim \boldsymbol{Y}^{(a)}$
    $\boldsymbol{Y}^{(a+1)} \leftarrow \boldsymbol{Y}^{(a)} - \boldsymbol{W}$
    $\boldsymbol{T} \leftarrow |\boldsymbol{W}| \times H$ array of zeros
    for $b = 1 \rightarrow B$ do
        $\boldsymbol{V} \leftarrow \{\}$
        for $j = 1 \rightarrow H$ do
            Sample $\boldsymbol{c}_{\boldsymbol{V},j} \sim \boldsymbol{c}_{\boldsymbol{U},j}$, where $|\boldsymbol{c}_{\boldsymbol{V},j}| = p$
            $\boldsymbol{V} \leftarrow \boldsymbol{V} \cup \boldsymbol{U}_{\boldsymbol{c}_{\boldsymbol{V},j}}$
        end
        $\boldsymbol{c}_{\boldsymbol{V}\cup\boldsymbol{W},1} \ldots \boldsymbol{c}_{\boldsymbol{V}\cup\boldsymbol{W},H} \leftarrow \text{Cluster}(\boldsymbol{V} \cup \boldsymbol{W})$
        for $i = 1 \rightarrow |\boldsymbol{W}|$ do
            $j = \underset{k \in \{1 \ldots H\}}{\arg\max} |\, \boldsymbol{c}_{\boldsymbol{V}\cup\boldsymbol{W},k} \cap \boldsymbol{b}_i|$
            $\boldsymbol{c}_{\boldsymbol{W},j} \leftarrow \boldsymbol{c}_{\boldsymbol{V}\cup\boldsymbol{W},j} - \boldsymbol{c}_{\boldsymbol{V},j}$
            $\boldsymbol{T}_{ih} = \boldsymbol{T}_{ih} + 1$, where $h = \underset{k \in \{1 \ldots H\}}{\arg\max} |\, \boldsymbol{c}_{\boldsymbol{V}\cup\boldsymbol{W},j} \cap \boldsymbol{c}_{\boldsymbol{V},k}|$
        end
        $\boldsymbol{U}^{(a+1)} \leftarrow \boldsymbol{U}^{(a)} \cup \boldsymbol{W}$
        for $i \leftarrow 1 \rightarrow |\boldsymbol{W}|$ do
            $\boldsymbol{c}_{\boldsymbol{U},j} \leftarrow \boldsymbol{c}_{\boldsymbol{U},j} \cup \boldsymbol{b}_i$ where $j = \underset{k \in 1 \ldots H}{\arg\max} \boldsymbol{T}_{ik}$
        end
        $a \leftarrow a + 1$
    end
end
for $j = 1 \rightarrow H$ do
    $\boldsymbol{r}_{c_{\boldsymbol{U},j}} = j$
end
// See Table 3.12 for a legend of the symbols in the algorithm
```

Algorithm 3: Streaming Consensus Clustering

Distributed Streaming Consensus Clustering

The streaming property of this algorithm permits finding the consensus clustering of an arbitrarily large input data set while running on a single commodity node, albeit at

Notation	Dimension	Meaning
$\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{W}$	$\lvert\boldsymbol{U}\rvert \times G, \lvert\boldsymbol{V}\rvert \times G, \lvert\boldsymbol{W}\rvert \times G$	Resolved, anchor, current set respectively
$\boldsymbol{c}_{\Gamma,j}$	$\lvert\boldsymbol{\Gamma}\rvert \times 1$	Indexes of elements in set $\Gamma \in \{\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{W}, \boldsymbol{V} \cup \boldsymbol{W}\}$ that are in cluster j
$\boldsymbol{T}$	$\lvert\boldsymbol{W}\rvert \times H$	Votes for each consensus cluster for each data point in $\boldsymbol{W}$
B	1 x 1	Number of resamples of $\boldsymbol{c}_{\boldsymbol{U},j}$ done for each $\boldsymbol{W}$

Table 3.12: Legend of symbols in Algorithm 3

the expense of time. In the event that there are more computational nodes available, the algorithm is also able to make use of those nodes and shorten wall-clock time. The key to doing this is to perform the initialization step in Section 3.2.2 on one master node and distribute both the initial resolved set, $\boldsymbol{U}^{(0)}$, and resulting clusters , $\boldsymbol{c}_{\boldsymbol{U},j}$, to all nodes to independently carry out the algorithm on their non-overlapping partitions of the remaining data. In this thesis, we combine both the distribution of the algorithm as well as streaming on the slaves to generate a consensus clustering on a dataset that is infeasible for MMM with Distributed EM given a particular amount of computing resources, achieving both time and accuracy results that are competitive with MMM with Distributed EM.

3.3 Results

In this section, we will discuss the two datasets that we have chosen to test both algorithms on, highlighting how each dataset tests the system in a different dimension. We will then present comparative results on the accuracy of the outcomes as well as the computational performance of the algorithms.

3.3.1 Datasets

Synthetic Dataset

We used the output of our random projection and clustering runs described in Section 2.5 as the input to the consensus clustering methods. Recall that this data was generated by projecting to a random subspace a dataset of 400 million points from an mixture of 10 8-dimensional Gaussian distributions (see Section 2.5.1). The projections were then clustered. The projection-clustering procedure was run 20 times to generate an input dataset of 400 million points by 20 base clusterings for the consensus clustering algorithms. Figure 2-5 shows the diversity of the base clusterings in terms of the normalized mutual information (NMI) between them.

We then ran both our MMM with Distributed EM and Streaming Consensus Clustering algorithms on 25, 50, 100 and 150 cloud nodes to understand how the performance of the algorithms changed with the number of nodes. Each cloud node was a virtual machine on an OpenStack cloud with 2 virtual CPUs and 4GB of memory. Each algorithm was run 10 times on each number of nodes setup to gain confidence and understand the variability in the results.

SUN 397

We also used a real dataset from the computer vision domain to demonstrate the ability of our methods to scale in other dimensions and handle real-world problems. We chose the SUN 397 (Xiao et al., 2010) dataset of 108,754 images where each image is labelled as one of 397 scene categories. Each category contains at least 100 images and it is a "well-sampled" subset of the larger SUN database which contains 131,072 images and 908 categories. A 2-level hierachy is built on the 397 categories which reduces the categories to 16 categories at the second level and 3 categories at the top level.

In choosing the dataset, we were looking for a dataset that was well matched to scene-based GIST features (Oliva and Torralba, 2001) since GIST feature generation code was readily available and easy to use. We therefore declined to use the 80 million images dataset (Torralba et al., 2008) nor ImageNet (Deng et al., 2009) neither of which is labelled by scenes despite them having more images than SUN397.

GIST features capture the "*spatial envelope*" of a scene which according to Oliva and Torralba (2001) may be described as "*a set of perceptual properties (naturalness, openness, roughness, ruggedness and expansion)*". Using code supplied by the authors, we generated 512 GIST features for each image and projected the features to a random subspace whose number of dimensions was governed by a sample drawn from the Gaussian distribution depicted in Figure 3-2. We centered this distribution at 70 dimensions and used a standard deviation of 25 in order to reduce the average runtime of each of the 10,000 GMM runs. We also farmed out the feature extraction and clustering tasks to a cloud of 100 nodes. For feature extraction each node handled a subset of images and for clustering, each node generated 100 clusterings. Both the MMM with Distributed EM and Streaming Consensus Clustering algorithms were run 10 times each on this dataset using 25 2-core, 4GB memory cloud nodes similar to those described in Section 3.3.1.

Table 3.13 compares both datasets on a number of characteristics. Given a fixed amount of computing resources, MMM with Distributed EM is unable to handle data sizes over a certain maximum. For instance, at 25 computing nodes with 4GB of memory each MMM with Distributed EM fails on the 400 million-point synthetic dataset. So for that configuration, we only present results from Streaming Consensus Clustering.

3.3.2 Accuracy Results and Comparison

In this section, we compare the accuracy of both algorithms on both datasets. The strengths of SCC are emphasized when it achieves comparable accuracy because as we shall see in later sections, it runs in less walk-clock time than the MMM with

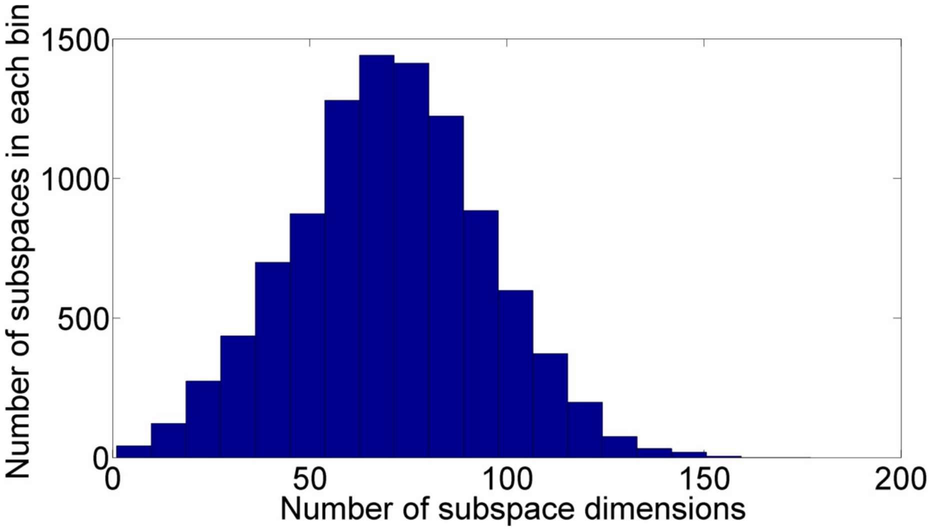


Figure 3-2: Distribution of random subspace dimensions GIST features were projected to

Distributed EM algorithm on the same number of computing nodes.

Synthetic Dataset Results and Comparison

We expected that the accuracy of the MMM with Distributed EM algorithm not to change much as the number of nodes changes since the same basic algorithm is running regardless of number of nodes. This is evident in Figure 3-3. Note that we were unable to run MMM with Distributed EM on 25 nodes due to memory limitations. With the Streaming Consensus Clustering algorithm, however, we expected that there would be a drop in the accuracy as the number of nodes increase. This is because, as the number of nodes increases, the size of the current set decreases and in a bid to maintain consistent mixture ratio between the current set and the anchor set, the size of the anchor set must be similarly reduced making it less representative of the full dataset. Interestingly, our results indicate that even at 150 nodes, the accuracy of the Streaming Consensus Clustering algorithm is not significantly lower than the accuracy of the MMM with Distributed EM method. This result may be sensitive to the dataset chosen.

Property	**Synthetic Dataset**	**SUN397 Dataset**
Number of data points (n)	400 million	90,331
Number of base clusterings (G)	20	10,000
Consensus clustering input data size	8 billion × 4 bytes = 29.8 GB	900 million × 4 bytes = 3.4 GB
Number of output clusters (m)	10	15
Number of parameters to be estimated	2010	2,250,015
Number of features before clustering	8	512
Minimum number of 4GB computing nodes required for MMM with Distributed EM (Leaving about 1GB for running the operating system)	40	2
Maximum amount of data that can be handled by MMM with Distributed EM on 25 4GB computing nodes	248 million points by 20 clusterings	2 million pictures by 10,000 clusterings or 90,331 images by 198,000 clusterings
Approximate size of data transfer during 1 MMM with Distributed EM iteration	1MB	1GB

Table 3.13: Comparison of two datasets used

SUN397

Achieving high accuracy was not the aim of introducing the SUN397 dataset since we were interested in a real world dataset that tested the *scalability* of the algorithms in another dimension: number of base clusterings. When compared to the ground truth, consensus labels were not very accurate according to either algorithm. Figure 3-4 shows that both MMM with Distributed EM and Streaming Consensus Clustering perform poorly with MMM with Distributed EM having higher accuracy on average than Streaming Consensus Clustering.

One possible reason for the accuracy difference between MMM-EM-CC and SCC is

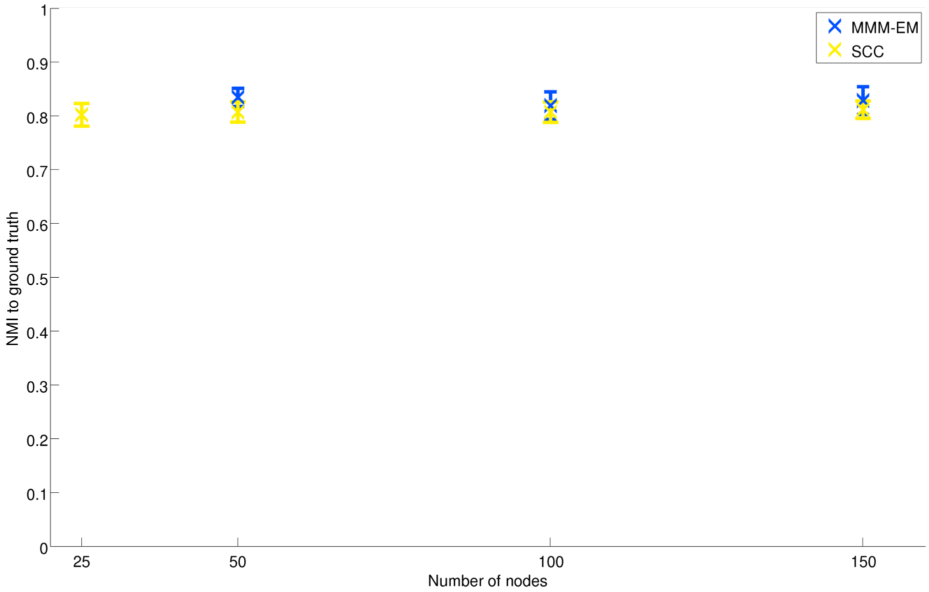


Figure 3-3: Average NMI to ground truth, over 10 runs, of two consensus clustering approaches on the synthetic dataset

that the SCC was not tuned for the nature of the SUN397 problem. This problem has fewer data points, leading to lower sizes of the anchor set and possible sampling issues.

We might be able to achieve better SCC results by changing the stopping criterion for the EM process on the slaves to perform more iterations. We could also do more anchor set resamplings per current set. These measures could potentially increase the SCC runtime, but performance results, shown later in this chapter, illustrate that MMM-EM-CC is an order of magnitude slower than SCC such that we could implement these measures without a damaging impact to the comparative time performance of SCC.

3.3.3 Wall-clock time

Synthetic Dataset

As expected, the results presented below indicate that the MMM with Distributed EM algorithm's performance does not take advantage of extra nodes at the same rate

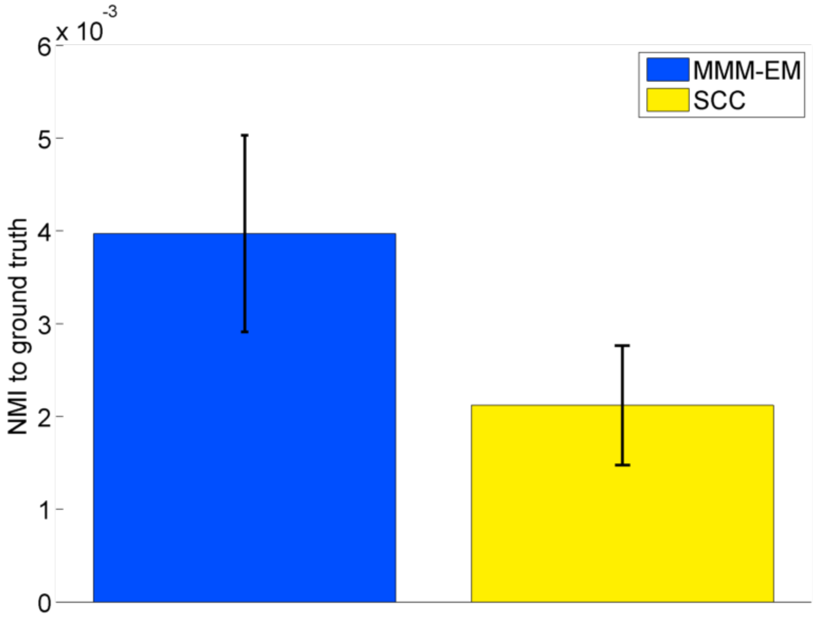


Figure 3-4: Average NMI to ground truth, over 10 runs, of two consensus clustering approaches on the SUN 397 dataset

that Streaming Consensus Clustering does. This is due to an I/O performance hit that the MMM with Distributed EM method takes with each extra slave. As more nodes are added, the steps in which the local parameters are aggregated to form the global parameters (three times in every iteration) becomes a bottleneck. If the network has latency or bandwidth issues or if the parameter size is increased as in the SUN397 problem, the performance hit becomes even more pronounced.

This is not a significant issue with Streaming Consensus Clustering because after the distribution of the initial anchor set to the slaves, the slaves do not communicate with the master till the end of the iteration. This higher efficiency is evident in Figure 3-5 which shows the average slave CPU utilization in both MMM with Distributed EM and Streaming Consensus Clustering. Note that the CPU utilization can be above 100% because some operations make use of both CPU cores on the slaves. MMM with Distributed EM shows decreasing CPU utilization with increasing number of nodes due to longer waits for global parameters. However, the speed gains from each node processing smaller partitions of the dataset as the number of nodes increase offset these bottlenecks and lead overall to shorter computation times per iteration and overall processing times (see Figures 3-6 and 3-7). We expect though, that at some high number of nodes and network latency, the marginal benefit of adding a node will be less than the performance hit.

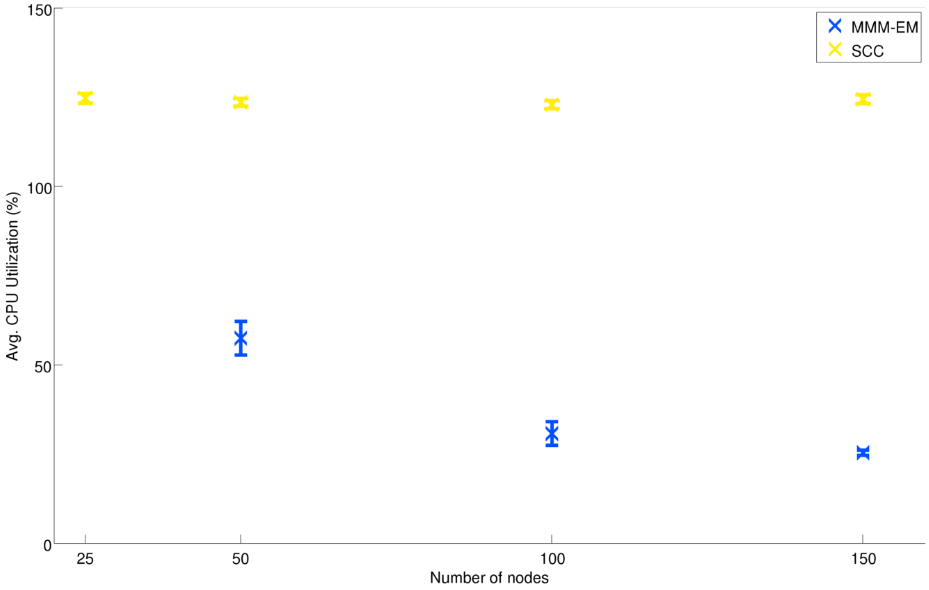


Figure 3-5: Average CPU utilization, over 10 runs, of two consensus clustering approaches on the synthetic dataset

Finally, note that SCC spends significantly less time per iteration than MMM-EM-CC because it incurs no communication cost. However, this does not directly translate to overall time savings in Figure 3-6 because 1) SCC generally executes more iterations due to resamplings and 2) the overall size of the data clustered being somewhat increased as a result of the anchor set i.e. many data points are clustered more than once: first as part of a current set and, possibly, as part of a later anchor set.. We, however, have more flexibility with the time cost of SCC because we are able to adjust the number of resamplings and the stopping criteria on the slaves to meet a time budget. We believe that SCC is robust to relaxing the slave stopping criteria because of the resamplings.

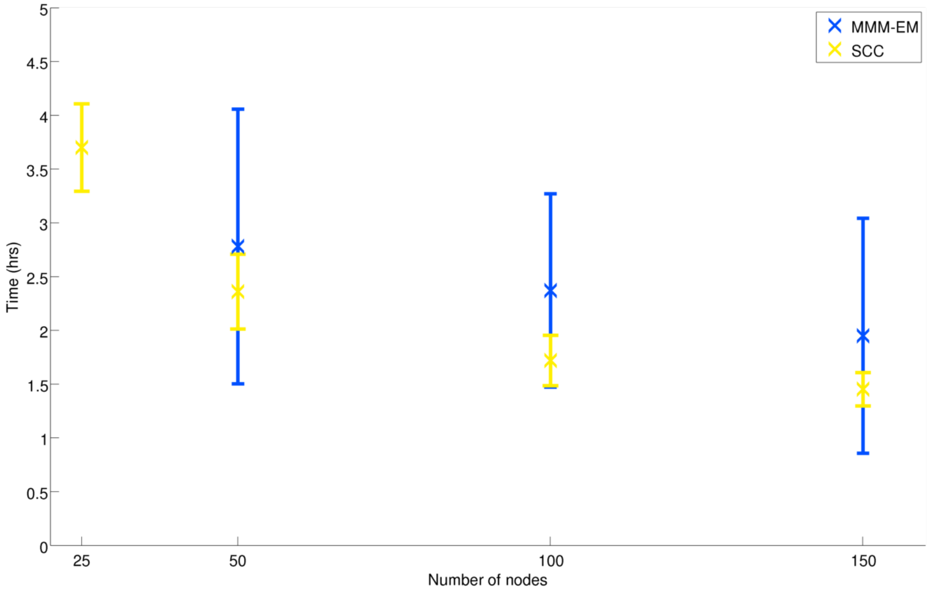


Figure 3-6: Average time taken per run, over 10 runs, of two consensus clustering approaches on the synthetic dataset

SUN397

The large number of clusterings in the SUN397 dataset and consequent large number of parameters emphasize the strengths of the SCC algorithm. As Figure 3-8 shows, MMM with Distributed EM takes 10x more time to complete clustering the dataset on 25 nodes. This is due to the large number of parameters that have to be transferred between each slave and the master once in every iteration. We observed the size of the transfer to be as large as 40MB per node per roundtrip per iteration which leads to an overall 20×25 nodes $\times 2$ (for roundtrip) $= 1$GB of data transferred in every iteration. Since SCC does not have such transfers, the algorithm completes much quicker than MMM with Distributed EM.

3.4 Conclusion

We demonstrated that the EM algorithm used for consensus clustering can be distributed to run on a large number of computing nodes by creating an implementation of this distribution scheme. This allowed us to empirically show the maximum limits

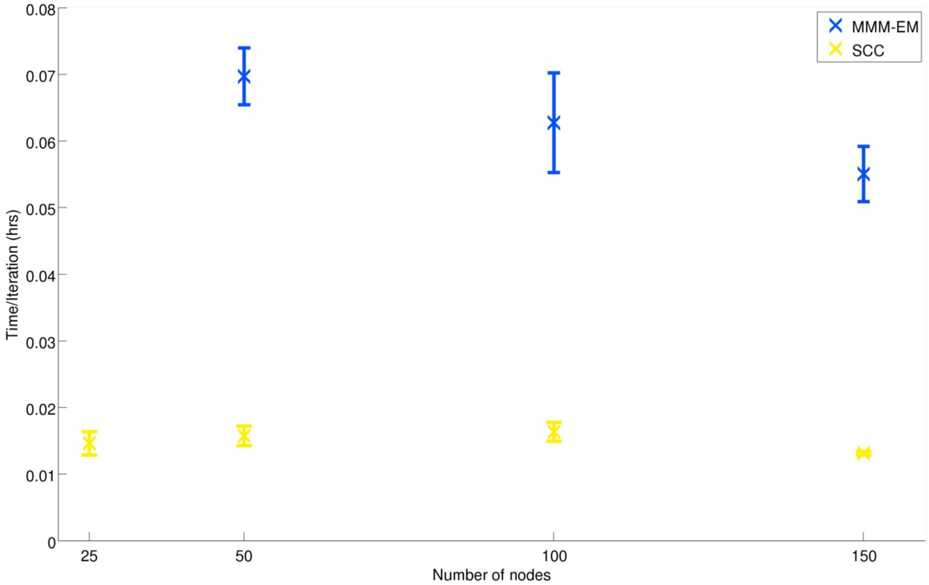


Figure 3-7: Average time taken per iteration, over 10 runs, of two consensus clustering approaches on the synthetic dataset

on the size of the dataset for a given number of nodes and memory size in each.

We also created an algorithm for streaming the consensus clustering operation without a limit on the size of the dataset it can handle as input in contrast to MMM with Distributed EM. We show empirically that this method achieves comparable accuracy results and will execute to completion more quickly by incurring smaller data transfers. By running both algorithms on two datasets that test the algorithms' ability to scale to a large number of nodes or a large number of base clusterings, we have demonstrated the robustness of both systems to a variety of problem types and scales.

3.5 Future work

The Streaming Consensus Clustering algorithm could be further sped up. We could implement a strategy allowing points in the current set to be added to the resolved set before all the resamples are completed. Currently, the current set is kept constant till all B resamples of the anchor set are completed. However, since the voting that

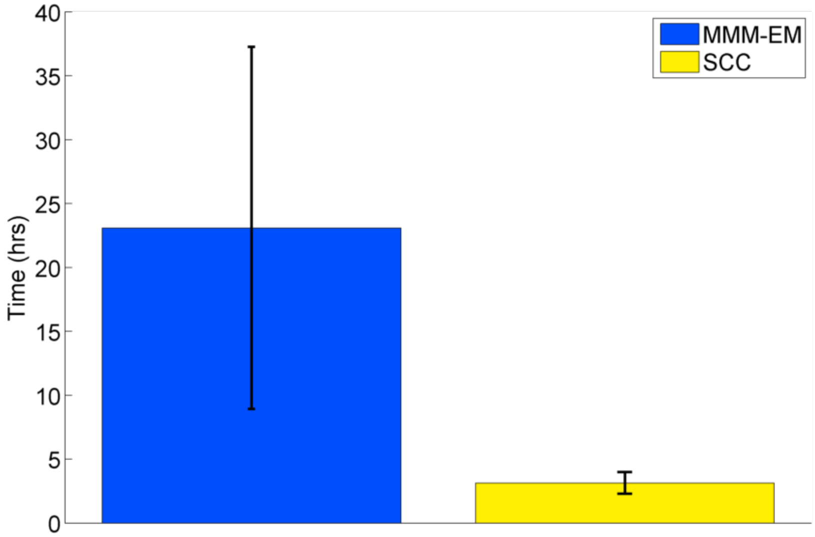


Figure 3-8: Average time taken per run, over 10 runs, of two consensus clustering approaches on the SUN397 dataset

is done on the current set is a simple majority vote, once any data point gets $> \frac{B}{2}$ votes for a particular consensus cluster, that point should be added to the resolved set and replaced with another point from the remaining unclustered dataset. This way, points will be added to the resolved set at up to twice the current rate leading to overall faster completion times, and there will be no impact on algorithm accuracy.

Chapter 4

Data Ownership

In Chapters 2 and 3 we presented data mining methods that are amenable to the type and scale of medical data that is available today and is increasingly being gathered in many modern healthcare facilities. However, for these methods to have any chance of improving the quality of care, they must be applied to actual patient data. Unfortunately, we fear that due to issues of data ownership, patient data is being siloed within hospital systems - hospitals are reluctant to aggregate this data among themselves or make it available for knowledge mining. In this chapter, we examine the issue of data ownership with a focus on ensuring the availability of large medical datasets for knowledge mining. We proceed in the following fashion. We:

- begin by defining certain terms that are essential to understanding our arguments
- justify our focus on medical data ownership by highlighting its possible impacts on data availability
- highlight the landmark cases that have shaped common law around data ownership
- survey the current statutory positions that different states in the United States have taken on medical data ownership, expanding on previous work by Stearns

(2000) and identifying four main classes of ownership stances

- make and justify a recommendation for the creation of Federal law for data ownership
- recommend the codification of co-ownership of medical data between the patient and healthcare provider.

We will now proceed to define some important terms used frequently in this chapter.

4.1 Definitions

Property rights: Honoré (1961) cited by Morales (2013) defined property rights as:

> the right to possess, the right to use, the right to manage, the right to the income of the thing, the right to the capital, the right to security, the rights or incidents of transmissibility and absence of term, the prohibition of harmful use, liability to execution, and the incident of residuarity.

Note that even though this definition does not include the right to exclude others from such property, some other definitions do (Balganesh, 2008). However, Morales (2013) points out that the right to exclude can be left out from the bundle of rights above without drastically affecting the sufficiency of the remaining rights. This point is important because in a later part of this chapter, we argue for co-ownership of medical data where one owner is not able to exclude the other from their intended uses.

Access: In contrast to ownership which is implied by property rights, *access* as it is typically used in this domain refers to the ability of a party to view and, perhaps, make copies of a medical record regardless of whether they are owners or not. The definition of property rights above imply access for the owner but we can foresee a few situations where owners do not have access. For example, patients can be denied access to their own mental records especially if the healthcare providers believe that access could be detrimental to the patient's well being.

Privacy: According to Earl (2008), privacy is the right of a patient to not have their medical information disseminated to the public.

Medical record: Morales (2013) defines a medical record as "*all written, typed, or electronically stored traces of any aspect of patient treatment that has official status within the hospital system*". There is a historically-created, subtle distinction between the record and the information on the record. The record is the physical media the information is stored on (e.g paper) while the information is the knowledge housed in the record. This distinction was relevant mostly before the advent of electronic health records (EHRs). They (EHRs), in being able to be displayed on read-only devices such as monitors or stored in multiple places, blur the distinction to a point where the difference becomes irrelevant. In this chapter, when we are conveying historical information, we retain the distinction. Otherwise, we simply use the term, "medical record", to reflect the reality of the present day.

4.2 Medical Data Ownership

Various stakeholders have argued that debating issues of medical data ownership is not as relevant as issues of access (Bloomrosen and Detmer, 2008; Safran et al., 2007) and, perhaps, privacy. We however insist on exploring this issue for two reasons:

1) Firstly, uncertain property rights are generally a bottleneck to exploiting such property since the uncertainty creates tensions between contestable owners. These tensions in turn prevent any party from fully engaging in transactions with that property. For instance, hospitals may be afraid of the legal fallout of selling or transferring patient data if it is not clearly theirs. In fact, Hall and Schulman (2009) state that these uncertain rights are slowing investments in electronic health

records (EHR) despite the HITECH Act's $20 billion incentive to promote EHR use.

2) Ownership questions have arisen in the past in court cases and will continue to arise as long as there is legal uncertainty in this area. As we shall show later, some of these cases have been used to form body of common law on the subject but the piecemeal legal landscape developed by common law is insufficient to address the uncertainties in this area. These uncertainties will continue to increase even as more types of medical data are gathered and these datasets become increasingly valuable.

Since data ownership is relevant to the subject of data availability, it is important to understand how the law currently defines ownership. The state of the law on the subject is contained within the body of common law as defined by a number of landmark cases. Some states in the US have also codified medical data ownership stipulations in their state statutes. In the next sections, we will therefore examine the cases that have defined common law, identifying the contribution each case brought to the body of common law. We will also identify which states have medical ownership stipulations and attempt to find the commonalities and differences between each state. In doing so, we will be able to highlight the flaws in the status quo and make recommendations for fixing them.

4.3 Common Law Medical Data Ownership

Over the years, common law has evolved partial answers to the question of medical data ownership. Each new case has brought with it, reinforcement of the past ones and, sometimes, an addition that takes care of a peculiarity that did not arise previously. In this section we examine how these answers have developed over time as a series of precedents. We therefore walk down the landmark cases in chronological order starting from the first known medical data ownership case in 1935. For each case, we present:

- A brief summary of the dispute before the court
- The judgement delivered
- The contribution of the decision to the body of common law

We outline five chosen cases because these five contributed something new to the body of common law on the subject. Some of the cases expanded the definition of what constitutes a medical record by differentiating the record from the information within it and, in some cases, defining the record to include x-rays, dental records, blood analyses etc. Other cases established the health care provider as the owner of the record and one case clarified that the allocation of property rights is currently not contained within the constitution. We will now delve into some more details.

4.3.1 McGarry v. J.A. Mercier, 1935

Dispute summary: In one of the first cases of medical data ownership McGarry v. J.A. Mercier Co. (1935), McGarry, a doctor had been treating one of Mercier Co.'s employees who had sustained an on-the-job injury. Mercier Co. sought the courts to compel McGarry to hand over x-ray films of the employee in question.

Court ruling: The court ruled that the films were the property of McGarry in the absence of a contrary agreement, and regardless of the fact that the physician was paid by for his services because:

1. The X-rays are practically meaningless to the layman
2. The X-rays constitute part of the doctor's record which in the aggregate have value "incident to a physician's ... experience"
3. The X-rays might constitute an important part of evidence in a malpractice case
4. Payers pay for medical services not records, records are a by-product of

the service and thus do not belong to the payer

Contribution: X-ray negatives as well as “microscopic slides of tissue” belong to the physician who obtained them in the process of delivering medical attention.

4.3.2 Wallace v. University Hospitals of Cleveland, 1959

Dispute summary: In Wallace v. University Hospitals of Cleveland (1959), Wallace, the plaintiff, sought an injunction to compel the University Hospitals of Cleveland to permit her attorney’s inspection of her hospital records as well as provide a copy of the records to her attorney. The hospitals defended their refusal of this request by stating, among other things, that the records were property of the hospital.

Court ruling: The court held that, even though the hospital owns the records since they constitute the administrative documents of the business, the patient still has a property right to the information in the records.

Contribution: This case made a distinction between a medical *record* and the *information* contained within it, demonstrating that property rights in one did not imply property rights in the other. The case also established that patients have a property right to the information in their medical records.

4.3.3 Bishop Clarkson Memorial Hospital v. Reserve Life Ins. Co., 1965

Dispute summary: The dispute in Bishop Clarkson Memorial Hospital v. Reserve Life Ins. Co. (1965) stemmed from the hospital’s refusal to let Reserve Life Insurance company inspect and make copies of, without a supervising physician, the medical records of its clients.

Court ruling: The court ruled that since the patient has a property right to the record, an authorised representative should be allowed to inspect and make copies of the record.

Contribution: In contrast to the patient's property right to the *information* in Wallace v. University Hospitals of Cleveland (1959), this case established that patient's have a property right to their medical *record* itself.

4.3.4 Matter of Culbertson, 1968

Dispute summary: In Matter of Culbertson (1968), Harold Culbertson, a physician, had requested that his executors burn the medical records of all his patients after his death. Culbertson's former patients who were the petitioners in this case tried to compel the executors to allow them inspect and make copies of the records.

Court ruling: The court ruled in favor of the petitioners, allowing them to inspect and make copies of their records before the records were destroyed.

Contribution: Part of the ruling of the court for the petitioners may be interpreted as an expansion of the definition of a medical record in McGarry v. J.A. Mercier Co. (1935). The implied definition includes, not just X-rays but, blood analyses, electro-cardiograms etc. A similar ruling in Matter of Striegel v. Tofano (1977) further extended this definition to include dental records as well. This case also reinforced that patients have a property right to their records.

4.3.5 Gotkin v. Miller, 1975

Dispute summary: In Gotkin v. Miller (1975) Jane Gotkin had been treated in a number of mental hospitals and sought to obtain her records for a book she was writing. The hospitals refused her request, and their position was supported by federal courts.

Ruling: Notably, the ruling pointed out that the 4th and 14th amendments of the US constitution which the Gotkins used to support their case did not provide property rights. Rather, those amendments enforced property rights already allocated by state law and common law. Since New York law at that time did not recognize a patient's property rights to their medical records, the courts dismissed that argument.

Contribution: The case made the first stipulation about the role that the Federal constitution plays in the data ownership issue: enforcement of rights defined elsewhere. The case also established that the property rights must be allocated by common law or state law.

The recognition in Gotkin v. Miller (1975) of the importance of state law in determining medical data property rights provides a strong motivation to examine the current status of these laws with the aim of identifying which states have laws and the similarities between their laws. We proceed to do this in the next section.

4.4 Medical Data Ownership in State Law

Table 4.1 shows the 16 states that have codified medical data ownership into their statutes or administrative code. It identifies four stances:

1. Nine states recognize the hospital as the owner. Most of the states that have a stipulation on medical data ownership fall in this category.

2. One state recognizes the physician as the record owner, South Carolina is the only state in this category.

3. One state, New Hampshire, that recognizes the patient as the owner of the information within the record. It is also worth noting that New Hampshire law is silent about ownership of the record itself and is the only state that makes the distinction between the record and the information contained within the record.

Stipulation	**States using**	**Relevant statutes**
Medical facility owns records	Alaska	7 Alaska Admin. Code 12.770
	Illinois	210 ILCS 85/6.17
	Maryland	COMAR 10.01.16.04
	Mississippi	Miss. Code Ann. §41-9-65
	North Carolina	10A N.C.A.C. 13B.3903
	Oregon	Or. Admin. R. 333-505-0050
	Pennsylvania	28 Pa. Code §115.28
	Tennessee	Tenn. Code Ann. §68-11-304
	Utah	U.A.C. R432-100-33
Physician owns records	South Carolina	S.C. Code Ann. §44-115-20
Information in record is patient's property	New Hampshire	RSA 332-I:1
Provider (hospital or doctor) owns records	Florida	Fla. Stat. §456.057
	Georgia	O.C.G.A. §31-33-3
	Louisiana	La. R.S. 40:1299.96
	Texas	22 TAC §165.1
	Virginia	Va. Code Ann. §54.1-2403.3

Table 4.1: Survey of states that have codified some stance on medical data ownership

4. Five states recognize the healthcare provider (i.e. hospital or doctor) as the owner. This group of states is mostly made up of states in the South East and their laws appear to be derivative of each other's. These laws are vague about whether the physician's or the hospital's property rights take precedence. The one exception is Virginia which, incidentally, is not in the South East. In Virginia, "*Medical records ... shall be the property of such health care provider or, in the case of a health care provider employed by another health care provider, the property of the employer.*" (Va. Code Ann. §54.1-2403.3)

The remaining 34 of 50 states do not have any explicit stance on data ownership. Such states will have to rely on common law when disputes arise. This is a problem because, as mentioned earlier, the uncertainties in common law legal landscape will dissuade some actors from making their medical datasets available. State laws are not immune to this uncertainty because the laws are inconsistent among themselves. This non-comprehensive and inconsistent legal landscape is the subject of the next section where we unify the key stances in common and state law, identify where they differ and further explain why this is a problem.

4.5 Issues with Data Ownership Legal Landscape

4.5.1 Issues with common law approaches

So far, common law answers to the medical data ownership question have been relatively consistent with each other:

- The physician owns medical records
- Medical records may include x-rays, blood analyses, electrocardiogram and dental records
- The patient has a property right in both the information and the record

However, common law is still not a sufficient legal stance because:

Common law is outdated: The first common law answer to the question of medical data ownership was ruled on in 1935. Between then and now, and especially in the last few years, a lot has changed in the area of medical data gathering and knowledge-mining thus making such old judgements less and less relevant. Unfortunately, this first case, McGarry v. J.A. Mercier Co. (1935), has been cited in a court decision as recently as 2010 (Holtkamp Trucking Co. v. Fletcher, 2010). The common law approach of gradually evolving the legal stance using more recent cases may not be sufficient either. The rapid changes in the area of medical data gathering requires a clean break from the past. This position is supported by Pound (1908):

> ...courts are less and less competent to formulate rules for new relations which require regulation. They have the experience of the past. But they do not have the facts of the present.

Common law is less democratic: Pound (1908) also forwards the notion that common law is generally less democratic than legislative approaches to issues because legislators are elected representatives of the people in contrast to the judges that make common law who are appointed to their positions. This is also true in the case of administrative rule-making which is typically used to add greater detail to laws legislated in the United States. In the Administrative Procedure Act which defines the due process for administrative rule-making, regulations are made after months of making public proposals and receiving, as well as responding to proposal comments from the general public. Rulemaking also frequently involves public hearings that permit debate on the proposed regulations. The judicial process hardly works this way and gives little room for the public to influence the decisions made.

4.5.2 The Shift to Electronic Records

The patchwork of common law and state statutes used to answer the data ownership question leads to significant differences and ambiguity from state to state and case to case. This inconsistency might have been manageable when paper records were primarily in use and as such were bound by state lines. However, with the advent and widespread adoption of electronic records, state lines have become irrelevant since data may easily be stored in a different state from the health service provider or possibly replicated to multiple data centers in different states to enable disaster recovery. The data centers, servers and databases that hold this data may also hold data from multiple health service providers domiciled in different states further complicating adherence to a mishmash of rules about ownership. To drive the point home, there are even states like Alabama that have codified how the rules of data management should be applied, albeit without codifying ownership rules: "*Licensure to practice medicine in Alabama determines treatment of medical data regardless of where the data is maintained*" (Code of Ala. §34-24-504)

4.5.3 Definition of a record

The differences that arise due to the inconsistency of the law on medical data ownership are not only found in how ownership is defined. There are also differences in what constitutes a medical record. As highlighted in Section 4.3, Gotkin v. Miller (1975) initialized the definition of a medical record to x-rays and tissue and Matter of Culbertson (1968) expanded it to include blood analyses and electro-cardiograms. In similar fashion, many of the states that have codified ownership have set out some definitions for what constitutes a record. Unfortunately, these definitions do not always agree with each other. For example, North Carolina excludes x-rays from the definition of a medical record (10A N.C.A.C. 13B.3903) whereas Pennsylvania specifically includes "radiology & radiotherapy" (28 Pa. Code §127.35). Such inconsistencies could easily cause legal conflicts in any process to aggregate medical data across state lines for research uses.

4.6 Recommendations

In this section, we propose two recommendations for deciding on the issue of data ownership. The first recommendation - enactment of Federal Law - is geared at fixing the problems with the status quo as identified in the previous section. We then proceed to recommend the form that such federal law should take by considering various possible stakeholders in the issue and coming up with a co-ownership proposal that would strike a balance between equity and efficiency.

4.6.1 Federal Law

The three issues with the status quo - the inadequacy of common law, the advent of electronic records and the differences in the definition of a medical record - have prompted the recommendations that will be discussed in this section. We, first of all, recommend that explicit law be created to mitigate the issues in common law approaches. Secondly, we recommend that this law should be federal in nature to deal with all the inconsistencies that arise in state law. While the merit in these recommendations is apparent given the flaws in the status quo, we further justify the need for a federal stance on the issue by demonstrating how medical data has become a matter of interstate commerce. We also demonstrate that, even within the medical domain, there is a precedent for this path from patchwork of common and state laws to federal law as seen in the Health Insurance Portability and Accountability Act thus emphasizing the feasibility of our recommendations. Finally, we argue that existing state laws on ownership have been motivated by privacy demands, not by the research goals that we target in this thesis and newly enacted laws can include such research motivations and have provisions that specifically make it easier to get access to data for research purposes.

Interstate Commerce

One of the prerequisites for the Federal government to wade into the issue and define laws that govern ownership is for the issue to border on interstate commerce. Healthcare information may not have been an issue of interstate commerce a few decades ago when hospitals kept their own paper records and patients rarely received healthcare services across multiple states. However, it can be argued that the present proliferation of hospital networks that cut across states and the massive adoption of electronic health records has made healthcare information an interstate issue. Additionally, as pointed out by Bishop (2002), patient data travels between insurance companies and health providers located in different states. This gives the Federal Government the right and responsibility to make policies for the sector. In the words of AHIMA (2003) speaking about medical privacy,

> Modern realities, including the movement of patients and their healthcare information across state lines, the exchange of such information through automated databases, and the emergence of multi-state providers, simply render anything less than federal standards impractical.

Finally, it is plausible that if the kind of data aggregation we advocate for ever happens, it would start at the level of hospital networks. According to American Hospital Association (2007), a hospital network is "*a group of hospitals, physicians, other providers, insurers and/or community agencies that work together to coordinate and deliver a broad spectrum of services to their community*", and there are over 400 hospital systems, networks and alliances in the United States (American Hospital Association, 2013). The largest ones include the Veterans Administration (VA), Healthcare Corporation of America (HCA) and Ascension Health (Dark Daily, 2009) and each of these three hospital networks all have one Electronic Health Records (EHR) system running or being deployed across all their facilities (Hammond et al., 2010; HCA, 2012; Rose, 2012). Having a single EHR deployed across the entire hospital network allows the kind of intra-system aggregated data analysis that we are developing methods for without having to deal with issues of integration and interoperability. Unfortunately, what we gain in technical interoperability is lost in legal non-interoperability due to the fragmentation of the legal landscape of medical data ownership. To illustrate how the fragmentation may affect a hospital network, Figure 4-1 shows the locations of HCA facilities (which all run one EHR system) overlaid on the legal stance of different states with respect to data ownership. This graphic shows that 42% of HCA's 302 facilities lie in the five states that define the provider (physician or facility) as the data owner, 34% in the nine states that define the facility as the owner, 3% in South Carolina that defines the physician as the owner, 1% in New Hampshire that defines the patient as the owner of the information in the records and the remaining 20% in the 34 states that have no stance on medical data ownership.

Following the HIPAA precedent

Before the enactment of the Health Insurance Portability and Accountability Act in 1996, many states had privacy laws (Cohen, 2006) governing the disclosure of private medical information. According to Hussong (2000), healthcare organizations

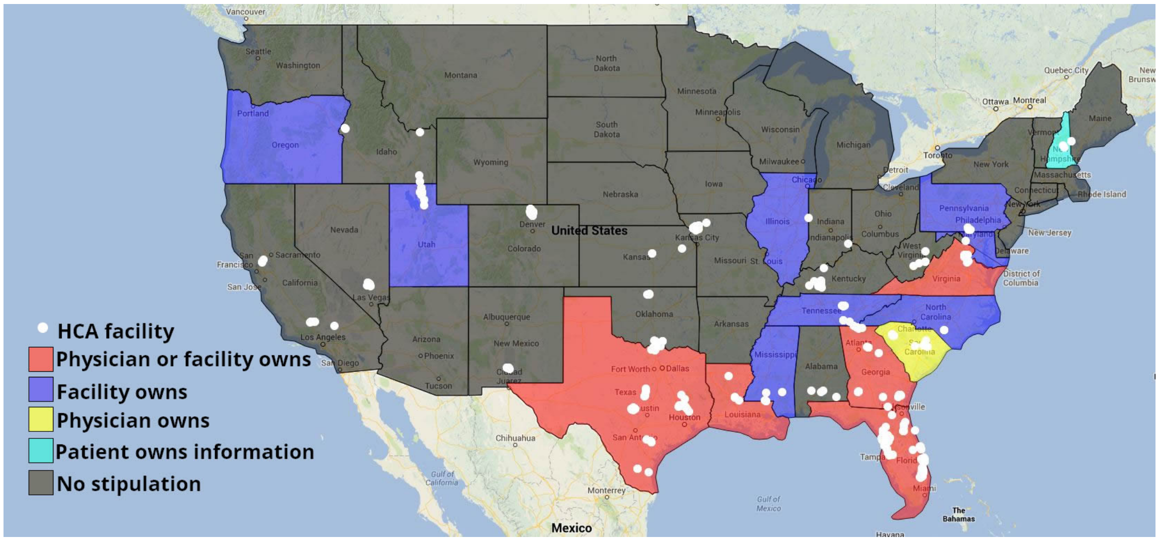


Figure 4-1: Locations of America's largest hospital network, HCA, also showing the host states' stance on medical data ownership

report more confusion and higher administrative costs with multiple state privacy laws. Federally-enacted HIPAA therefore brought some legal uniformity to the issue of privacy which was good given the increasing importance of electronic health records and the issues associated with privacy law fragmentation. To deal with the existing state laws, HIPAA includes provisions that allows it pre-empt state laws that are contrary to it (Cohen, 2006). We therefore argue that since the issue of data ownership is in a similar situation to where privacy was, pre-HIPAA, it is only logical that the Federal law precedent set by HIPAA be followed for medical data ownership.

Addressing the research motivation

After studying the existing state statutes on ownership as well as common law on the subject, it is our conclusion that these legal stances are motivated by a need to protect a patient's privacy and right of access. This is evidenced by subtle points in the various states' statute, for instance, the Illinois ownership stipulation is located in a section titled "*Protection of and confidential access to medical records and information*"(210 ILCS 85/6.17). The laws are also targeted at protecting the rights of healthcare providers with regards to their business records. However, in this thesis we are concerned with a different motivation for addressing the ownership question - access to data for research. A new (Federal) law concerning ownership of medical data will be able to add this increasingly important issue to the current motivations and address the niceties that may not be fully addressed when medical data mining was less prominent or practical.

4.6.2 Allocation of Property Rights

Even if the proposition of enacting Federal law to allocate medical data property rights is accepted, the question still remains of who the property rights in the data should be allocated to. There are a number of actors that can be considered here: government, patients, healthcare providers (facilities and physicians), insurance companies and device manufacturers. A case may be made for each of these actors to have a property right to the data. However, we dismiss the insurance companies and device manufacturers immediately. Even though the insurance company pays, they pay for treatment and not the medical record. Hence, they have no true stake to the data provided they get sufficient access for their business operations. Conversely the device manufacturers, since the hospital pays for the equipment, cannot lay claim to the data gathered by it. We will now give some more focus to the remaining players: the government, patient and healthcare provider.

Government

The government has been proposed by some as a possible owner of medical data. Specifically, Rodwin (2009) proposes that the government set up a central public repository that healthcare providers will be compelled by Federal law to submit anonymized patient data to. He argues that treating patient data as private property will preclude the formation of aggregated databases. While there may be some merit to this argument, we have also seen that private hospital networks are deploying EHRs across their entire network and these networks could easily be the precursors to a national medical database given that sufficient work is done in standardization of access to the network databases.

A similar point Rodwin (2009) tries to make is that private ownership of patient data will exclude public uses. However, Evans (2011) rebuts this by pointing out that the government still has the power to invoke takings on private property. This rebuttal is supported by an analogy with the private takings of land used for building rail tracks. We also add that the government has, in the past, shown a preference for private ownership for things like government-funded research. This was enabled by the Bayh-Dole Act of 1980 which, as The Economist (2002) highlights, led to rapid diffusion into *public* uses of research that would have otherwise gathered dust. The Bayh-Dole Act example also demonstrates the limited ability of the government to optimize the use of public resources - a fate that could be shared by patient data if ownership is assigned to the government. Additionally, citing the Privacy Act (5 U.S.C. §552a), Evans (2011) points out that, in federal government hands, patient data may be subjected to even more privacy requirements which will make research access more difficult. The Privacy Act requires federal agencies to notify individuals with personally-identifiable information before such information is disclosed.

The Patient

Allocating exclusive property rights to the patient might sound like the equitable thing to do but it might not be the most efficient. Lessons from intellectual property in synthetic biology have demonstrated that fragmented property rights could lead to an "*anticommons*" problem (Heller and Eisenberg, 1998). Heller and Eisenberg (1998) discusses how, in synthetic biology, fragmented intellectual property rights have made it increasingly difficult for scientists to innovate given the amount of intellectual property (IP) required in every project and the difficulties and transaction costs associated with negotiating license agreements with all the IP owners. Similarly, Evans (2008) compared such dispersion of property rights to the physical rights of way that have to be acquired at high transaction costs for infrastructure development.

Allocating medical data property rights to the patients runs the risk of creating such an anticommons problem where data consumers will find it infeasible to negotiate with all the data owners.

Another reason put forward (by some stakeholders) for patient ownership is for privacy protection. However, this is more an outcome of conflating a number of patient data-related issue: privacy, access, control and ownership. This is partly due to the interrelationships between these issues. For instance, if privacy were strictly enforced such that, say, only the patient and the provider who made the record have access to it, it would preclude other stakeholders e.g. another doctor from the access they need to work effectively. However, these issues can be dealt with separately without using ownership as a proxy for privacy. For instance, HIPAA addresses many privacy pain points without touching on the ownership question and the ideas we propose in this thesis should do the same for ownership without negatively impacting the patient's privacy. This stance is shared by Rodwin (2009).

Finally, if a patient is made the sole owner of their record and the ownership right spills into the domain of access and control, patients may be able to require that their approval is sought before their data is used in any form for research. Evans (2011) points out that this may pose a threat to the validity of the research results because even if the researchers are able to reach every patient in the system some patients would not approve of the use and this will possibly introduce a statistical bias in the sample obtained.

Nonetheless, we remain mindful of the patient's right to access the data and use it as they see fit. We therefore propose a system where a patient is an owner but not the sole owner of the data. The patients' co-ownership rights will therefore not exclude other owners of the data (the healthcare provider) from using it provided privacy is protected and access for all necessary parties is preserved.

Healthcare Provider

We believe that having the healthcare provider own the data, as it is currently defined in some states, strikes the right balance between equity and efficiency. It is an equitable approach given the effort the hospital puts into making and maintaining the records. It is also efficient because it does not disperse the rights to so many parties as to risk falling into Heller and Eisenberg (1998)'s "tragedy of the anticommons". This efficiency driven by aggregation of rights is even more so because of the existence of hospital networks and network-wide EHR systems as discussed earlier.

The stipulation in Virginia provides an ideal model for such a law because it clearly gives a hospital superior property rights in the records than the physicians employed there thus further curtailing the fragmentation of those rights:

> Medical records maintained by any health care provider ... shall be the property of such health care provider or, in the case of a health care provider employed by another health care provider, the property of the employer. (Va. Code Ann. §54.1-2403.3)

4.7 Future Work

One direction this work could be expanded to would be to look outside the United States for countries implementing either the changes we have proposed or some alternative policy stance and to study the effect on data availability in those countries. This research direction will also have to account for differences in legal, cultural and commercial systems in the countries studied.

Chapter 5

Conclusion

In this thesis we set out to address some of the technical and policy challenges to effective mining of the massive amounts of medical data currently being gathered in and outside hospitals. Our main contributions were:

- Our implementation of a distributed EM algorithm for Gaussian mixture models demonstrates a method for initializing the parameters of the algorithm when data is streamed in without having to store the entire dataset in memory or know the size of the dataset a priori. We also present a partial parameter update method that allows the distributed EM operation continue when a computing node holds up an iteration step. These capabilities will enable medical data mining at very large data scales on modest quantities of commodity computers.

- One contribution in the area of consensus clustering is our implementation of a distributed EM algorithm for performing consensus clustering using multinomial mixture models. This implementation, to the best of our knowledge, is the only such distributed implementation of a similar single-node algorithm created by Topchy et al. (2004). Our implementation allowed us demonstrate the inability of the algorithm to handle datasets past a certain size for a given amount of computational resources. Specifically, the algorithm was unable to process a 400 million-point dataset (30GB) on a cloud of 25 computing nodes with 4GB of memory each.

- Our main contribution to the consensus clustering body of knowledge is an algorithm for scaling any consensus clustering method to be able to handle large amounts of data that would have neither fit in memory of a single computing node nor even a cloud of nodes running the MMM with Distributed EM algorithm. Our algorithm has the following advantages:

  - It can perform consensus clustering on any amount of data given any amount of computing resources
  - It can perform consensus clustering on data that is streamed in
  - It is demonstrably faster than the MMM with Distributed EM algorithm

  We ran this algorithm on as few as 25 computing nodes with 4GB of memory, processing 400 million data points (30GB) in less than 4 hours and achieved normalized mutual information of about 0.8 which is about the same average received with MMM with Distributed EM when run on a sufficient number of computing nodes.

- Finally, we described the legal position of the United States on the issue of medical data ownership as defined in common law and state law. We then recommended that explicit law be created to replace the outdated common law. We also recommended that the law be federally defined to deal with the non-comprehensiveness and inconsistency of state law. Our final recommendation is that property rights in the data be allocated to both the patient and the healthcare provider.

Bibliography

Abdi, H., Valentin, D., Chollet, S., and Chrea, C. (2007). Analyzing assessors and products in sorting tasks: Distatis, theory and applications. *Food quality and preference*, 18(4):627–640.

AHIMA (2003). Confidentiality of Medical Records: A Situation Analysis and AHIMA's Position. Available http://library.ahima.org/xpedio/groups/public/documents/ahima/bok2_000623.hcsp, accessed June 2013.

American Hospital Association (2007). Fast facts on US hospitals. Available http://www.aha.org/aha/resource-center/Statistics-and-Studies/fastfacts.html, accessed June 2013.

American Hospital Association (2013). American Hospital Association Guide.

Balganesh, S. (2008). Demystifying the right to exclude: Of property, inviolability, and automatic injunctions. *Harv. JL & Pub. Pol'y*, 31:593.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*, volume 4. Springer, New York.

Bishop, R. H. (2002). Final Patient Privacy Regulations under the Health Insurance Portability and Accountability Act - Promoting Patient Privacy or Public Confusion? *Ga. L. Rev.*, 37:723.

Bishop Clarkson Memorial Hospital v. Reserve Life Ins. Co. (1965). 350 F. 2d 1006. Court of Appeals, 8th Circuit.

Bloomrosen, M. and Detmer, D. (2008). Advancing the framework: Use of health dataa report of a working conference of the american medical informatics association. *Journal of the American Medical Informatics Association*, 15(6):715–722.

Blumenthal, D. (2010). Launching HITECH. *New England Journal of Medicine*, 362(5):382–385.

Brill, S. (2013). Bitter pill: Why medical bills are killing us. Available http://www.time.com/time/magazine/article/0,9171,2136864,00.html accessed March 2013.

Chitta, R., Jin, R., Havens, T., and Jain, A. (2011). Approximate kernel k-means: Solution to large scale kernel clustering. In *Proc. ACM SIGKDD*, pages 551–556.

Cohen, B. (2006). Reconciling the HIPAA privacy rule with state laws regulating ex parte interviews of plaintiffs treating physicians: a guide to performing HIPAA preemption analysis. *Houston Law Review*, 43(1091).

Dark Daily (2009). Nations list of top ten largest healthcare systems include some surprises. Available http://www.darkdaily.com/nations-list-of-top-ten-largest-healthcare-systems-include-some-surprises-113, accessed June 2013.

Dasgupta, S. (2000). Experiments with random projection. In *Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence*, pages 143–151. Morgan Kaufmann Publishers Inc.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 1–38.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE.

Dhillon, I. S., Guan, Y., and Kulis, B. (2004). Kernel k-means: spectral clustering and normalized cuts. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 551–556. ACM.

Earl, M. (2008). Concise dictionary of modern medicine, by joseph c. segen. *MEDICAL REFERENCE SERVICES QUARTERLY*, 27(1):121.

Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. Kdd.

Evans, B. J. (2008). Congress' new infrastructural model of medical privacy. *Notre Dame L. Rev.*, 84:585.

Evans, B. J. (2011). Much ado about data ownership. *Harv. JL & Tech.*, 25:69.

Fern, X. Z. and Brodley, C. E. (2003). Random projection for high dimensional data clustering: A cluster ensemble approach. In *Proceedings of the Twentieth International Conference on Machine Learning*, volume 20, page 186.

Fred, A. L. and Jain, A. K. (2002). Data clustering using evidence accumulation. In *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, volume 4, pages 276–280. IEEE.

Gotkin v. Miller (1975). 514 F. 2d 125. Court of Appeals, 2nd Circuit.

Gu, D. (2008). Distributed EM algorithm for Gaussian mixtures in sensor networks. *Neural Networks, IEEE Transactions on*, 19(7):1154–1166.

Guha, S., Rastogi, R., and Shim, K. (1998). Cure: an efficient clustering algorithm for large databases. In *ACM SIGMOD Record*, volume 27, pages 73–84. ACM.

Hall, M. A. and Schulman, K. A. (2009). Ownership of medical information. *Journal of the American Medical Association*, 301(12):1282–1284.

Hammond, K. W., Efthimiadis, E. N., Weir, C. R., Embi, P. J., Thielke, S. M., Laundry, R. M., and Hedeen, A. (2010). Initial steps toward validating and measuring the quality of computerized provider documentation. In *AMIA Annual Symposium Proceedings*, volume 2010, page 271. American Medical Informatics Association.

HCA (2012). Frequently Asked Questions about hCare. Available http://hcare.acareerathca.com/hca-ehr-faq/, accessed June 2013.

Heller, M. A. and Eisenberg, R. S. (1998). Can patents deter innovation? The anticommons in biomedical research. *Science*, 280(5364):698–701.

Holtkamp Trucking Co. v. Fletcher (2010). Ne 2d.

Honoré, A. M. (1961). Ownership. *Oxford essays in jurisprudence*, 107:107–28.

Hussong, S. J. (2000). Medical records and your privacy: developing federal legislation to protect patient privacy rights. *Am. JL & Med.*, 26:453.

Lin, X., Clifton, C., and Zhu, M. (2005). Privacy-preserving clustering with distributed EM mixture modeling. *Knowledge and Information Systems*, 8(1):68–81.

Matter of Culbertson (1968). 57 Misc. 2d 391. NY: Surrogate's Court, Erie.

Matter of Striegel v. Tofano (1977). 92 Misc. 2d 113. NY: Supreme Court, Orange.

McGarry v. J.A. Mercier Co. (1935). 272 Mich. 501; 262 N.W. 296. Supreme Court of Michigan.

Milenković, A., Otto, C., and Jovanov, E. (2006). Wireless sensor networks for personal health monitoring: Issues and an implementation. *Computer communications*, 29(13):2521–2533.

Morales, F. J. (2013). The property matrix: An analytical tool to answer the question," is this property?". *U. Pa. L. Rev.*, 161:1125–1125.

Oliva, A. and Torralba, A. (2001). Modeling the shape of the scene: A holistic representation of the spatial envelope. *International journal of computer vision*, 42(3):145–175.

Pound, R. (1908). Common law and legislation. *Harvard Law Review*, 21(6):383–407.

Rodwin, M. A. (2009). The case for public ownership of patient data. *JAMA: The Journal of the American Medical Association*, 302(1):86–88.

Rose, J. (2012). Evidence-driven quality improvement, the Ascension way. How one large hospital system spreads perfomance improvement throughout its enterprise.

Interview by Mark Hagland. *Healthcare informatics: the business magazine for information and communication systems*, 29(4):26.

Safran, C., Bloomrosen, M., Hammond, E., Labkoff, S., Markel-Fox, S., Tang, P. C., and Detmer, D. E. (2007). Toward a National Framework for the Secondary Use of Health Data: An American Medical Informatics Association White Paper. *Journal of the American Medical Informatics Association*, 14(1):1–9.

Schoen, C., Osborn, R., Huynh, P. T., Doty, M., Zapert, K., Peugh, J., Davis, K., et al. (2005). Taking the pulse of health care systems: experiences of patients with health problems in six countries. *Health Affairs*, 24:5.

Smith, D. M. and Williams, M. (2010). Data loss and hard drive failure: Understanding the causes and costs. Available http://www.deepspar.com/wp-data-loss.html, accessed June 2013.

Stearns, P. V. (2000). A comparison of state laws: Access to and cost of reproduction of patient medical records. *The Journal of Legal Medicine*, 21:79–108.

Strehl, A. and Ghosh, J. (2003). Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *The Journal of Machine Learning Research*, 3:583–617.

The Economist (2002). Innovation's golden goose. *The Economist*, 12.

Topchy, A., Jain, A. K., and Punch, W. (2004). A mixture model of clustering ensembles. In *Proc. SIAM Intl. Conf. on Data Mining*.

Torralba, A., Fergus, R., and Freeman, W. T. (2008). 80 million tiny images: A large data set for nonparametric object and scene recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(11):1958–1970.

Vlassis, N. and Likas, A. (2002). A greedy EM algorithm for Gaussian mixture learning. *Neural Processing Letters*, 15(1):77–87.

Wallace v. University Hospitals of Cleveland (1959). 164 NE 2d 917.

Welling, M. and Kurihara, K. (2006). Bayesian k-means as a "maximization-expectation" algorithm. In *Sixth SIAM International Conference on Data Mining*, volume 22.

Wolfe, J., Haghighi, A., and Klein, D. (2008). Fully distributed EM for very large datasets. In *Proceedings of the 25th international conference on Machine learning*, pages 1184–1191. ACM.

Xiao, J., Hays, J., Ehinger, K. A., Oliva, A., and Torralba, A. (2010). Sun database: Large-scale scene recognition from abbey to zoo. In *Computer vision and pattern recognition (CVPR), 2010 IEEE conference on*, pages 3485–3492. IEEE.

Xu, R., Wunsch, D., et al. (2005). Survey of clustering algorithms. *Neural Networks, IEEE Transactions on*, 16(3):645–678.

Zhang, T., Ramakrishnan, R., and Livny, M. (1996). Birch: an efficient data clustering method for very large databases. In *ACM SIGMOD Record*, volume 25, pages 103–114. ACM.