Semi-Supervised Learning and Generalized Mixture Models in the analysis of Retinal Images

Samuel J. Frame *†
S. Rao Jammalamadaka*
Department of Statistics and Applied Probability
University of California, Santa Barbara

Spring 2006

Abstract

The Bio-Image Informatics research program at UCSB is a collaborative effort funded by the National Science Foundation. The statistical computing problem we address in this paper is to tailor an application of the Generalized Mixture Models (GMM's) for analyzing biological images, and the development of versatile software that is useful for this purpose. This analysis has several goals. First, we demonstrate that such methods can be used as objective diagnostic tools for classifying new images in the medical and biological context, instead of relying on subjective human analysis. Second, we are interested in using GMM's to better understand the similarities and differences between various classes of controlled experiments. We do this by inspecting and learning from fitted model components. Further, we are able to test for the equality of local regions (represented by model components) from different known classes. After a brief introduction of the GMM's, we discuss a case-study that has been of interest to biologists in this project.

*Supported by the National Science Foundation, ITR-03316979
†Supported by NASA's California Space Grant Graduate Fellowship Fund

1 Introduction

During the past several years, a large number of researchers have accumulated libraries of raw information in the form of digital images. In particular, researchers at the University of California Santa Barbara's (UCSB's) Neurosciences Research Institute (NRI) are involved in research projects that result in a large number of biological digital images and related experimental data. Due to the controlled nature of these experiments, much is known about the biological images and what experimental conditions they resulted from. What one would like to know, however, is the relationship between the cellular/sub-cellular activities captured in these images and a clearer understanding of the physiological mechanisms underlying such systems.

Also during the last few years, Generalized Mixture Models (GMMs) and related Semi-Supervised Statistical Learning methods have been shown to be very useful in data mining, analysis, and classification problems in highly populated image reservoirs [9] [10]. In all cases, the image collections consist of images where the ground-truths are either completely known (possibly via experimental control or visual inspection by an expert analyst) or images where ground-truths are unknown and wish to be inferred. GMMs can be used to classify unlabeled images, understand the differences between various classes, and formally test for statistical differences between the classes.

In this part of the ITR project, we focus on the retinal detachment problem. We use GMMs and semi-supervised learning to find multivariate mixtures which best capture similarities and dissimilarities between biological images such as retinal images collected under different experimental conditions.

2 Generalized Mixture Models

Semi-supervised learning methods have become increasingly popular over the last 10 years [5]. Prior to semi-supervised methods, much of the work in the statistics community has focused on either *unsupervised* or *supervised* learning methods until recently [11]. Much of the literature on the subject exists in the computer science and engineering fields [15].

Unsupervised learning typically refers to clustering unlabeled data points (either in low or high dimensions) into homogenous groups [5] [1] [12] [7], i.e. all the data points have class (or group) labels which are unknown or hidden. Unsupervised learning methods seek to cluster or group these data points as specified by user controlled model parameters (such as the number of components in a standard mixture model). Methods for unsupervised learning include the standard mixture model and aglormorate/divisive k-means hierarchical clustering methods [5] [1]. These methods are essentially fancy exploratory data analysis tools. That is, the grouping structure found has to be explicitly and carefully inspected by users/analysts to gleam valuable information about the groups that are found. The problem is that it is unknown precisely how many distinct groups have be found and how many there actually are. For example, consider an unsupervised mixture model with 2 known classes that selects a mixture model with 4 components to best represent the probability density function of the unlabeled data (say using BIC to choose the number of components). What did the model find exactly? Did it find 2 components for each class or 3 components for one class and 1 component for the other? In such cases, visual and analytic inspection of the mixture components can lead to interesting analysis and grouping structures after such an ad hoc analysis. On the opposite extreme, the method may have found a clustering decision which reveals little about the true grouping structure of the data.

On the opposite end of the spectrum are fully supervised learning methods [5] [12]. These

methods use observations where the class or grouping structure is known to the user. In such cases where the grouping/class structure is known, supervised learning methods seek to use this information to find a model that best separates these known classes, subject to some criteria like for instance minimizing the cross-validated prediction errors) [5]. Standard supervised statistical learning methods include standard mixture models, LDA/QDA methods, and methods such as generalized linear models and generalized additive models. Such methods are very useful for exploring the separation between distinct known classes, testing the significance of such separations, and using these schemes for classifying future observations which are believed to have come from one of the known classes. Although supplementary methods are available for detecting unknown classes, fully supervised learning methods fail to detect unknown classes that are not present in training the supervised model. Semi-supervised learning methods bridge this gap quite elegantly.

Semi-supervised statistical learning methods utilize both labeled and unlabeled data points. For an excellent review and survey of literature see [15] and for a more recent reference, [9]. This merges the unsupervised learning (purely unlabeled data) with supervised learning (purely labeled data) together to form a model which uses both sets of data simultaneously. Specific motivation for this statistical framework comes from data situations such as the retinal image problems. Without loss of generality, motivation for semi-supervised statistical learning methods, as applied to Generalized Mixture Models, comes from data situations where large amounts of both labeled and unlabeled data are available. From the labeled data, supervised learning is needed to find a model which best separates the known classes whereas unsupervised methods are needed to learn about the unlabeled data to learn about outliers regions in known classes and the possible unknown classes itself. Taken together, semi-supervised learning methods are capable of simultaneously finding models which separate known classes and are able to detect unknown/anomolous data points. The benefit of this method, in contrast to most semi-supervised learning methods, is the concept of unknown class discovery and the ability to make *explicit* unknown class inference. In fact, this is the first inference made for unlabeled samples. Given a known class inference, the secondary classification inference is to decide which known class each unlabeled samples belong to.

The remainder of this section focuses on the Generalized Mixture Model and related semi-supervised learning method. In Section 3.1, we introduce some notation and model formulation. Section 3.2 outlines the Expectation Maximization, semi-supervised learning method and Section 3.3 details inferential methods for classification and testing.

2.1 Notation and Model Formulation

Semi-supervised learning entails using both labeled and unlabeled data. As such, we define our data X, to be a combination of both labeled and unlabeled data. The data is denoted by $X = \{X_l, X_u\}$ where there are N_l labeled samples (l for labeled) $X_l = \{(\underline{x}_1, l, c_1), ..., (\underline{x}_{N_l}, l, c_{N_l})\}$ and $(N - N_l)$ unlabeled data points $X_u = \{(\underline{x}_{N_l+1}, m), ..., (\underline{x}_N, m)\}$ with the label absence indicator m representing the fact that the class labels are missing/hidden and, hence, unlabeled.

Notice that the data incorporates the additional information which indicates the presence/absence of a class label $L \in \{l, m\}$. Under this data formulation, GMMs explain all of the observed data including the label presence/absence information. Generalized mixture models differ from standard mixture models in that GMMs explicitly use labeled and unlabeled data and that GMMs explicitly seek to model and explain the additional label presence/absence information by way of model formulation. The method by which GMMs explain and model this information is by allowing for different types of mixture components which differ in how they generate labeled and/or unlabeled data points.

1. *Predefined Components:* These generate data samples which are both labeled and unlabeled where we assume that the data labels are *missing at random* [**?**] [9]. These components will exclusively represent known classes. Note that more than 1 component can represent a single known class.

2. *Non-predefined Components:* These generate data samples which are exclusively unlabeled. As such, these mixture components will represent outlier regions of known classes or unknown classes.

As mentioned earlier, the benefit of GMMs is the ability to discover an unknown class(es) (should the learning algorithm deem the existence of one necessary to better explain the observed data structure) and having an explicit inferential method to determine if unlabeled samples belong to such an unknown class. The ability to make unknown class inference, in addition to the label presence/absence information explained by the model via defining the 2 different types of mixture components, separates GMMs from standard mixture models. Standard mixture models do not use both labeled an unlabeled data, they do not seek to explain the label presence/absence, nor are they capable of explicitly making unknown class inference by way of model formulation.

Let M be the number of mixture components in a GMM and let M_k denote k^{th} mixture component for $k = 1, ..., M$. Let C_{pre} denote the subset of components are which are *predefined* components with remaining subset of *non-predefined* components denoted by $\overline{C}_{pre}$. The mechanism by which GMMs explain the label presence/absence information is by probabilistically quantifying the rate at which a generic, predefined component will generate labeled data i.e. $P(L = l|M_g \in C_{pre})$ where M_g represents a generic, predefined mixture component. Note that this probability is the same for all predefined components such that this probability is "tied across all components" which are predefined (extended models exist which allow for the label/absence probability to be specific to each class or component). Further, since non-predefined components exclusively generate unlabeled data, $P(L = l|M_g \in \overline{C}_{pre}) = 0$.

For class representation, let P_c denote the set of all known classes with $c(\underline{x}) \in P_c$ denoting a class label from a sample $\underline{x}$ which originates from a know class. Another appealing aspect of GMMs is the probabilistic (or soft) ownership of classes by components. Recall that predefined components explicitly represent the known classes, i.e. those that exist in P_c. This representation is probabilistic in the sense that each predefined component has a mass function over the predefined or known classes denoted by $P(C = c|M_k \in C_{pre}), \forall M_k \in C_{pre}, c \in P_c$ and, much more specific to the probabilistic association between components a specific sample $\underline{x}$ from a specific class $c(\underline{x})$ we have $P(C = c(\underline{x})|M_k \in C_{pre}, L = l)$. Each component will have a component-weight denoted by α_k with general density function $f(\underline{x}|\underline{\theta}_k)$.

Let
$$v_k = \begin{cases} 1 & \text{if } M_k \in C_{pre} \\ 0 & \text{if } M_k \in \overline{C}_{pre} \end{cases}$$
such that the dummy variable v_k is the explicit, mathematical mechanism which distinguishes predefined and non-predefined components. The $\{v_k\}s$ are the so called "component natures" which detail the nature of each component as either predefined or non-predefined. With this notation, we are able to state the joint data log-likelihood of the observed data for a model with M components

as

$$\begin{aligned}\log L_M &= \sum_{\underline{x}\in X_l} \log(\sum_{k=1}^{M} v_k \alpha_k f(\underline{x}|\underline{\theta}_k) P(L=l|M_g \in C_{pre}) P(C=c(\underline{x})|M_k \in C_{pre}, L=l)) \\ &+ \sum_{\underline{x}\in X_u} \log(\sum_{k=1}^{M} v_k \alpha_k f(\underline{x}|\underline{\theta}_k) P(L=m|M_g \in C_{pre}) + (1-v_k)\alpha_k f(\underline{x}|\underline{\theta}_k)). \quad (1)\end{aligned}$$

Given suitable amounts of data as well as the number of components M (later work treats M a model parameter with efforts to try to estimate it explicitly), the parameters which must be learned in (3) are:

$$\begin{aligned}\Lambda &= \{\{\alpha_k\}_{k=1}^M, \{\underline{\theta}_k\}_{k=1}^M, P(L="l"|M_g \in C_{pre}), \{P(C=c|M_k, L="l"), \forall c \in P_c\}_{M_k \in C_{pre}}, \{v_k\}_{k=1}^M\} \\ &= \{\Lambda_{EM}, \{v_k\}\}.\end{aligned}$$

We note briefly that the current implementation of this semi-supervised learning method uses a multi-stage Expectation Maximization (EM) algorithm for parameter estimation . Model selection is done via Bayesian Information Criteria (BIC) and component testing uses the well known multivariate T^2 test procedure [12].

2.2 Semi-supervised Learning and Model Selection

In this section, we describe the current method (also as a matter of numerical implementation) of estimating the model parameters in addition to estimating the number of components in the model, M. For fixed M, we use a generalized Expectation Maximization (EM) algorithm [7] [1] [13] [5]. The generalized EM algorithm consists of 2 steps: (i) choose the component natures, the $\{v_k\}$'s, to maximize (3) given all other parameters are held fixed and then (ii) use EM to estimate the remaining parameters given the component natures are held fixed. As with EM, we are guaranteed to have nondecreasing $\log L_M$. However, EM does not always guarantee convergence to global optima [9] [7] [5].

Estimating the Component Natures
Depending on the size of the model as indicated by M, there are 2 ways to choose the component natures. If M is not too large, then one can enumerate all possible 2^M combinations of the component natures (each component either 0 or 1) and select the combination which maximizes $\log L_M$ in (3). For large M, this strategy grows exponentially with M and is simply not feasible. The sub-optimal alternative (yet still having the property of have a non-decreasing $\log L_m$) is detailed in [9] [10] which is an iterative "one at a time" selection of the natures. That is, we cycle through the components and choosing a single v_k one at a time, while holding all others fixed, by considering the $\log L_M$ "score" associated with the values v_k can take on, either $\{0, 1\}$. The value of the single v_k which maximizes $\log L_M$ is selected. This is done for all the $\{v_k\}$'s and this cyclic choosing is repeated until no more changes are made. Although no convergence global is guaranteed, the $\log L_M$ is guaranteed to be none-decreasing.

EM for the Remaining Model Parameters
Expectation Maximization (EM) is the standard frequentist, maximum likelihood estimation (MLE) method for estimating the parameters of a mixture model [1] [7] [5]. Due to computational advances in the last 20 years, its value beyond theoretical existence has been proven for difficult estimation

problems such as those when one wishes to estimate the parameters of a mixture model [**?**] [10]. Applying EM estimation methods for Generalized Mixture Models is not different with exception estimating and updating a number of additional parameters namely $P(L = l|M_g \in C_{pre})$ and $P(C = c|M_k, L = l)\ \forall c \in P_c$.

Without loss of generality, the EM algorithm treats the observed data as incomplete in the sense that the component which generates the observation is unknown. Assuming the existence of such a grouping indicator variable, it is possible to define a log likelihood associated with the complete set of data denoted $\log L_C$. EM uses the $\log L_C$ in 2 distinct steps :

1. *Expectation* (E-step)
   Take the expected value of the complete log likelihood $E[\log L_C|\Lambda^{(t)}]$ given the current set of parameter estimates is held fixed. Such an expectation yields an equation with the expectation of a grouping indicator variable for each sample. This step is the "ownership" step where we seek to find the probability of component owning a data point (for all the data points).

2. *Maximization* (M-step)
   Given the probabilistic ownership of components owning samples, the M-step finds parameter estimates for the remaining parameters in the the model by finding estimates which maximize $E[\log L_C|\Lambda^{(t)}]$.

Let
$V_{\underline{x}k} = \begin{cases} 1 & \text{if } \underline{x} \in M_k \\ 0 & \text{otherwise} \end{cases}$
so that $V_{\underline{x}k}$ is the indicator variable detailing which component the data point $\underline{x}$ originates from, namely component M_k in this case. It is easy to show that the associated complete log likelihood is

$$\begin{aligned} \log L_C &= \sum_{\underline{x}\in X_l}\sum_{k=1}^{M} v_k V_{\underline{x}k} \log(\alpha_k f(\underline{x}|\underline{\theta}_k) P(L = l|M_g \in C_{pre}) P(C = c(\underline{x})|M_k \in C_{pre}, L = l)) \\ &+ \sum_{\underline{x}\in X_u}\sum_{k=1}^{M} v_k V_{\underline{x}k} \log(\alpha_k f(\underline{x}|\underline{\theta}_k) P(L = m|M_g \in C_{pre})) \\ &+ \sum_{\underline{x}\in X_u}\sum_{k=1}^{M} (1 - v_k) V_{\underline{x}k} \log(\alpha_k f(\underline{x}|\underline{\theta}_k)). \end{aligned} \tag{2}$$

E-step
More precisely in the E-step, we have the expectation of our grouping variable $V_{\underline{x}k}$ given the current set of parameter estimates $\Lambda^{(t)} = \{\Lambda^{(t)}_{EM}, \{v_k\}\}$ is held fixed. Since $V_{\underline{x}k}$ is binary, in $E[\log L_C|\Lambda^{(t)}]$ we have

$$\begin{aligned} E[V_{\underline{x}k}|\underline{x} \in X_l, \Lambda^{(t)}] &= 1 \cdot P(V_{\underline{x}k} = 1|\underline{x} \in X_l, \Lambda^{(t)}) + 0 \cdot P(V_{\underline{x}k} = 0|\underline{x} \in X_l, \Lambda^{(t)}) \\ &= P(V_{\underline{x}k} = 1|\underline{x} \in X_l, \Lambda^{(t)}) \\ E[V_{\underline{x}k}|\underline{x} \in X_u, \Lambda^{(t)}] &= P(V_{\underline{x}k} = 1|\underline{x} \in X_u, \Lambda^{(t)}) \text{ follows similarly.} \end{aligned}$$

It is easy to show with Bayes rule that these probabilities are

$$P(M_k|\underline{x} \in X_l, \Lambda^{(t)}) = \begin{cases} \dfrac{\alpha_k f(\underline{x}|\underline{\theta}_k) P(C = c(\underline{x})|M_k \in C_{pre}, L = l)}{\sum_{k' \in C_{pre}} \alpha_{k'} f(\underline{x}|\underline{\theta}_{k'}) P(C = c(\underline{x})|M_{k'} \in C_{pre}, L = l)} & \text{if } M_k \in C_{pre} \\ 0 & \text{if } M_k \in \overline{C}_{pre} \end{cases} \tag{3}$$

and

$$P(M_k|\underline{x} \in X_u, \Lambda^{(t)}) = \begin{cases} \frac{\alpha_k f(\underline{x}|\underline{\theta}_k) P(L = m"|M_g \in C_{pre})}{\sum_{k' \in C_{pre}} \alpha_{k'} f(\underline{x}|\underline{\theta}_{k'}) P(L = m|M_g \in C_{pre}) + \sum_{k' \in \overline{C}_{pre}} \alpha_{k'} f(\underline{x}|\underline{\theta}_{k'})} & \text{if } M_k \in C_{pre} \\ \frac{\alpha_k f(\underline{x}|\underline{\theta}_k)}{\sum_{k' \in C_{pre}} \alpha_{k'} f(\underline{x}|\underline{\theta}_{k'}) + \sum_{k' \in \overline{C}_{pre}} \alpha_{k'} f(\underline{x}|\underline{\theta}_{k'})} & \text{if } M_k \in \overline{C}_{pre} \end{cases} \quad (4)$$

so that the expected complete log likelihood is

$$\begin{aligned} E[\log L_C|\Lambda^{(t)}] &= \sum_{\underline{x} \in X_l, k \in C_{pre}} P(M_k|\underline{x} \in X_l, \Lambda^{(t)}) \log(\alpha_k f(\underline{x}|\underline{\theta}_k) P(L = l|M_g \in C_{pre}) P(C = c(\underline{x})|M_k \in C_{pre}, L = l)) \\ &+ \sum_{\underline{x} \in X_u, k \in C_{pre}} P(M_k|\underline{x} \in X_u, \Lambda^{(t)}) \log(\alpha_k f(\underline{x}|\underline{\theta}_k) P(L = m|M_g \in C_{pre})) \\ &+ \sum_{\underline{x} \in X_u, k \in \overline{C}_{pre}} P(M_k|\underline{x} \in X_u, \Lambda^{(t)}) \log(\alpha_k f(\underline{x}|\underline{\theta}_k)). \end{aligned} \quad (5)$$

M-step

To best demonstrate the M-step, we assume that $f(\underline{x}|\underline{\theta}_k)$ is a Multivariate Normal Distribution with mean vector and covariance matrix $(\underline{\mu}_k, \Sigma_k) = \underline{\theta}_k$ respectively. For all of the components in a model with M components, we are left to solve $E[\log L_C|\Lambda^{(t)}]$ for the values of the remaining parameters $\Lambda_{EM}^{(t+1)} = \{\{\alpha_k\}, \{\underline{\theta}_k\}, P(L = l|M_g \in C_{pre}), \{P(C|M_k, L = l)\}\}$ which maximize $E[\log L_C|\Lambda^{(t)}]$ which yields for each component $k = 1, ..., M$:

$$\alpha_k^{(t+1)} = \frac{\sum_{\underline{x} \in X_l} P(M_k|\underline{x} \in X_l, \Lambda^{(t)}) + \sum_{\underline{x} \in X_u} P(M_k|\underline{x} \in X_l, \Lambda^{(t)})}{N} \quad (6)$$

$$\underline{\mu}_k^{(t+1)} = \frac{\sum_{\underline{x} \in X_l} \underline{x} P(M_k|\underline{x} \in X_l, \Lambda^{(t)}) + \sum_{\underline{x} \in X_u} \underline{x} P(M_k|\underline{x} \in X_l, \Lambda^{(t)})}{\sum_{\underline{x} \in X_l} P(M_k|\underline{x} \in X_l, \Lambda^{(t)}) + \sum_{\underline{x} \in X_u} P(M_k|\underline{x} \in X_l, \Lambda^{(t)})} \quad (7)$$

$$\Sigma_k^{(t+1)} = \frac{\sum_{\underline{x} \in X_l} (\underline{x} - \underline{\mu}_k^{(t+1)})(\underline{x} - \underline{\mu}_k^{(t+1)})^T P(M_k|\underline{x} \in X_l, \Lambda^{(t)}) + \sum_{\underline{x} \in X_u} (\underline{x} - \underline{\mu}_k^{(t+1)})(\underline{x} - \underline{\mu}_k^{(t+1)})^T P(M_k|\underline{x} \in X_l, \Lambda^{(t)})}{\sum_{\underline{x} \in X_l} P(M_k|\underline{x} \in X_l, \Lambda^{(t)}) + \sum_{\underline{x} \in X_u} P(M_k|\underline{x} \in X_l, \Lambda^{(t)})} \quad (8)$$

$$P(C = c|M_k \in C_{pre}, L = l) = \frac{\sum_{\underline{x} \in X_l : c(\underline{x}) = c} P(M_k|\underline{x} \in X_l, \Lambda^{(t)})}{\sum_{\underline{x} \in X_l} P(M_k|\underline{x} \in X_l, \Lambda^{(t)})} \quad \forall c \in P_c \quad (9)$$

$$P(L = l|M_g \in C_{pre}) = \frac{\sum_{\underline{x} \in X_l} \sum_{k \in C_{pre}} P(M_k|\underline{x} \in X_l, \Lambda^{(t)})}{\sum_{\underline{x} \in X_l} \sum_{k \in C_{pre}} P(M_k|\underline{x} \in X_l, \Lambda^{(t)}) + \sum_{\underline{x} \in X_u} \sum_{k \in C_{pre}} P(M_k|\underline{x} \in X_u, \Lambda^{(t)})}. \quad (10)$$

The general outline of the Semi-supervised Learning method is:

1. Learn the $\{v_k\}$ via cycling through them one at a time picking the individual value of v_k which maximizes $\log L_M$ and repeating this process until no further changes are made. Denote this set of updated or learned parameters as $\{v_k\}^{(t+1)}$.
2. Use $\Lambda^{(t)} = \{\Lambda_{EM}^{(t)}, \{v_k\}^{(t+1)}\}$ and do EM learning until sufficient convergence has been achieved. Denote this updated or learned set of parameters as $\Lambda^{(t+1)} = \{\Lambda_{EM}^{(t+1)}, \{v_k\}^{(t+1)}\}$.

Model Selection
Up until this point, we have developed GMMs based on the assumption that the number of components, M, is known. In this section we briefly describe how we estimate the number of components in the mixture model, M. When fitting a standard mixture model with a learning method such the generalized EM algorithm, the standard method by which M is selected is Bayesian Information Criteria (BIC) [9] [**?**] [7] [5].

$$BIC = \frac{1}{2}\log(N)\sum_{k=1}^{M} P_k - \log L_M \tag{11}$$

where N is the number of data points, P_M is the number of parameters completely specifying component k, and $\log L_M$ is the log likelihood of a model specified by M components.

Computationally, choosing M this way is extremely inefficient and time consuming. It requires that, for each value of M, models are extensively learned with the pre-described semi-supervised learning method. To relieve computative burdens, M should be upper restricted by the number of components which are supported by the size of the data. Lower restricting the number of components is to assume that each known class should be represented by at least one component and consider at least one potential component for a possible unknown class.

Even with such restrictions, the number of possible models to explore can still be numerically overwhelming. Miller makes further suggestions such as "overestimating" the number of components and then reducing the model size by a single component and considering the effect on the log likelihood of the model [9] [5]. Methods such as these, which are better than nothing, still have problematic issues such as how to best/optimally choose the component to eliminate (and thus, redistribute the ownership of data points and associated updating of model parameters). There are several methods to achieve this goal, yet none of them result in reduced models which are "subsets" of the original larger model. Alas, these methods are not conducive to generalized likelihood comparison methods and may be unreliable. Such methods rely on BIC evaluations to determine if the reduced model is better.

We are currently considering fully Bayesian methods of estimating the number of mixture components using Markov Chain Monte Carlo methods[2] [4] [14]. They use methods which change the number of components by adding or subtracting a single component via birth/death respectively. The benefit of these methods is they do not require exhaustively searching the entire parameter space of the model dimension.

2.3 Inference and Testing

We now pay attention to the merit of GMMs for the purposes of inference and describe how we can use multivariate statistical hypothesis tests to evaluate and understand the mixture classes that are found. We will statistically test for component and class similarities and differences. This will lead to checking each component of the model in order to identify significant and homogeneous biological and cellular events, in the case of the ITR project.

Classification
GMMs can make 2 levels of classification. First, they can help predict if an unlabeled sample belongs to a known or unknown class, and (ii) given a known class, they can be used to predict which known class the unlabeled sample comes from.

The *MAP* probability that an unlabeled sample belongs to an unknown class is given by

$$P(M_g \in \overline{C}_{pre} | \underline{x} \in X_u) = 1 - \sum_{k \in C_{pre}} P(M_k | \underline{x} \in X_u, \Lambda^{(t)}) \tag{12}$$

where $P(M_k | \underline{x} \in X_u, \Lambda^{(t)})$ is given in equation (6) as the component ownership of unlabeled sample $\underline{x}$. Values of $P(M_g \in \overline{C}_{pre} | \underline{x} \in X_u)$ greater than .5 suggest that the sample originates from an unknown class.

Given we have made a *known class* inference for an unlabeled sample $\underline{x}$, the *a posteriori* known class probability is given by

$$P(C = c | \underline{x} \in X_u, \Lambda^{(t)}) = \frac{\sum_{k \in C_{pre}} \alpha_k f(\underline{x} | \underline{\theta}_k) P(C = c | M_k \in C_{pre}, L = l)}{\sum_{k \in C_{pre}} \alpha_k f(\underline{x} | \underline{\theta}_k)}, \forall c \in P_c. \tag{13}$$

Samples, then, are then assigned to the class for which this *MAP* is the largest.

Testing
In GMM model selection, we assume that all of the components selected in the model learning process are *significantly* different from one another. As we will see, however, there will exist instances where mixture components are not as significantly different from one another. Furthermore and oddly enough, we will see that components which are **not** significantly different from one another may actually come from radically different classes. This may suggest that there are local regions within 2 different classes that are quite similar to one another and, further, they are more like each other than the original classes to which they belong.

To test similarities between classes, we will employ a multivariate T-test for testing the equality of mean vectors i.e. test

$$\begin{aligned} H_0 &: \underline{\mu}_k = \underline{\mu}_{k'} \\ H_1 &: \underline{\mu}_k \neq \underline{\mu}_{k'}. \end{aligned} \tag{14}$$

Since we use multivariate Normal distributions, we need the component mean and covariance as the "sample" mean and covariance $\overline{x}, S$ respectively. One question to address is, how to estimate the sample size of each of the smaller classes?

Ideally, we would look at the component weight, α_k, and take the proportion of samples (both labeled and unlabeled) which that component owns namely $n_k = \alpha_k * N$ where N is the total sample size. The weights outlined in (8) would achieve this. However, the numerical implementation of the semi-supervised mixture model algorithm does not reflect this "true" meaning/value of the $\{\alpha_k\}$'s. The algorithm is implemented to *preserve class mass* which, as it sounds, gives equal weight to each class by preserving mass/weight of each component (within each class) relative to the total number of classes (this may include a presumed unknown class) [10]. In this sense, the component's mass/weight reflects the amount of data from a particular class which is owned by that component.

For example, if a single component represents a single class in a 3 class problem (perhaps 2 known classes and 1 unknown class) then its weight would always be $\frac{1}{3}$ reflecting that it owns all of the data from that class. Note that, this is problematic and clearly not consistent with the outlined theory. Alternative versions of the implementation are being developed to correct for this inconsistency.

For now, we will estimate the number of samples each component owns empirically. That is, we look at the total number of samples which each component owns as indicated by the $P(M_k|\underline{x} \in X_l, \Lambda^{(t)})$ and $P(M_k|\underline{x} \in X_u, \Lambda^{(t)})$ mass functions by

$$\begin{aligned} n_k &= \frac{1}{N}[\sum_{x \in X_l} I(P(M_k|\underline{x} \in X_l, \Lambda^{(t)}) = max(P(M_k|\underline{x} \in X_l, \Lambda^{(t)}))) \\ &+ \sum_{x \in X_u} I(P(M_k|\underline{x} \in X_u, \Lambda^{(t)}) = max(P(M_k|\underline{x} \in X_u, \Lambda^{(t)})))]. \end{aligned}$$

Once we have the necessary sample mean vectors, covariance matrices, and component sizes we conduct Hotelling's T^2 test [12]. Let p be the number of elements of the mean vector, $\underline{\mu}_k$, for testing components k_1 and k_2.

Let

$$S_p = \frac{1}{n_{k_1} + n_{k_2} - 2}((n_{k_1} - 1)S_{k_1} + (n_{k_2} - 1)S_{k_2})$$

which gives way to the intuitive test statistic T^2 defined by

$$T^2 = \frac{n_{k_1} n_{k_2}}{n_{k_1} + n_{k_2}}(\underline{\mu}_{k_1} - \underline{\mu}_{k_2})^T S_p^{-1}(\underline{\mu}_{k_1} - \underline{\mu}_{k_2})$$

which can be easily transformed into a statistic with a convenient distribution

$$T_F^2 = \frac{n_{k_1} + n_{k_2} - p - 1}{p(n_{k_1} + n_{k_2} - 2)} T^2$$

which has an F distribution with p numerator degrees of freedom and $n_{k_1} + n_{k_2} - p - 1$ denominator degrees of freedom [12].

Note that this is for testing a single hypothesis. All pairwise hypothesis can be tested similarly by simply adjusting the rejection region or associated P-Values to account for the multiple hypothesis tests being tested [12].

3 Retinal Image Informatics Application

One goal of the ITR project is to study the effects of retinal detachment, reattachment, and any treatments that can be used. Retinal images are images of a cross section or slice of a retina from a subject, obtained with confocal microscopy or some other device. Generally, the subjects used in these experiments are cats, mice, rats, dogs, and monkeys. Non-human subjects are used because some of the experimental procedures so traumatize the subjects that they must be euthanized after the data is collected. Generally, retinal images are gathered in experiments in the following way:

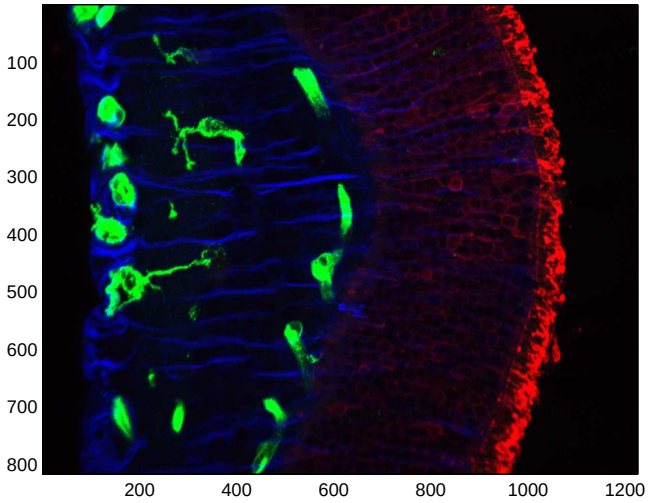

Figure 1: A retinal image taken 3 days after detachment

1. Trauma
   Subjects selected for the non-control regime of the experiment are subjected to a traumatic event causing damage to areas and tissues of interest in the retina. Since retinal detachment is the primary event of interest, separation of the retina is done manually or with a chemical solution which causes sufficient damage effectively causing retinal detachment. The detachment can serve as a reasonable paradigm for degenerative diseases for which the effect is the same.

2. Experimental Conditions
   Non-control subjects are then subjected to one of the several experimental conditions after detachment. This may include the use of a drug or treatment (such as placing the damaged retina in an oxygen solution) or allowing the retina to be separated for a pre-determined length of time. Included as an experimental condition is retinal reattachment where detached retinas are re-attached for various lengths of time and under treatment conditions.

3. Collection
   After the retinas undergo experimental conditions, they are imaged using a biological imaging equipment such as a confocal microscope. Since the retina is a 3-dimensional object, often cross section slices are imaged one at a time. The standard retinal image, shown eg. in Figure 1, is an image of a cross section of a retina. The *metadata* collected includes information about the type of subject (cat, mouse, etc), the length of retinal detachment and, if available, the length of retinal reattachment , the antibodies/dyes used to stain the tissue for imaging purposes, the aspect of the image (whole mount as opposed to a specific section), and the experimental conditions.

Thus, the data available are the cross-section slices of imaged retinas and the associated metadata (experiment info, subject info, ect). Since these images are gathered under a highly controlled environment, using GMMs to predict class labels (as defined by length of detachment/reattachment and experimental conditions) is not of primary concern here since this informa-

tion is already known. Moreover, there does not exist any unlabeled data for which classification is needed (however unlabeled data does exist in the context of unlabeled testing data which is to be used in the learning process).

What is of scientific interest is the differences between various classes and whether or not these differences are statistically and biologically significant. Additionally, in terms of a classification problem, we wish to understand the reasons (biological or statistical) that images are misclassified during cross validation. Our approach helps analyze and understand the effects of retinal detachment with the following procedure:

1. Generalized Mixture Models
   We use GMMs to fit a mixture of multivariate normal distributions to each of the retinal image classes, as defined by experimental conditions and length of detachment/re-attachment. In this sample analysis the data comes from 2 known classes namely: 3 days of reattachment (3D), versus 3 days of detachment + 28 days of reattachment (3D+28R).

2. Classification
   We test the separation of the various classes via cross validated classification. We then inspect and analyze the misclassified samples to determine the nature of the misclassified and its biological significance.

3. Testing
   We use multivariate statistical methods to test the equality of mixture model components in local regions of the same class and across components from different classes. The outcome of these tests will be particularly useful for biological interpretation and conclusion.

Since we use the Generalized Mixture Model as the engine for these analyses, we now present a general, theoretical description of Generalized Mixture Models and the associated Semi-Supervised Statistical Learning method.

We will use GMM's to model, analyze, classify, and explore various classes of retinal images using 2 sets of features developed by the ITR research program. We will begin by briefly describing the data that we use in the analysis, the classes we use, and how many samples are available for each class. Then, we will describe how we use GMMs computationally and give a generic model fitting overview. The first set of features we use are statistical summaries which we will develop and interpret. The second, and most recent set of features developed, are Gabor Filter, Texture Histogram features.

3.1 The Data

The data for this analysis are cross-sectional biological images taken of the feline retina with confocal microscopy imaging methods. For this analysis, we use 4 distinct classes which appear even from a casual visual inspection, different from one another. The 4 classes are defined by the length of detachment and any reattachment. The classes are

Normal: images gathered under normal conditions for which there are 29 samples

3D: images gathered after 3 Days of retinal detachment for which there are 15 samples

3D+28R: images gathered after 3 Days of retinal detachment followed by 28 Days of re-attachment

for which there are 46 samples

7D: images gathered after 3 Days of retinal detachment for which there are 27 samples.

Note that these sample sizes are quite small relative to what would be desirable.

3.2 The GMM Setup

Cross validation techniques are required to fairly and accurately evaluate the GMMs inference abilities [7] [5]. To do so completely, we use K-fold cross validation [13] [5]. This cross validation method "removes" K observations (for each class) and considers them to be *unlabeled* where the remaining observations are considered to be *labeled*. For each class, mixture model components are initialized and "learned" (using EM learning) as predefined and associated with their respective class with probability of nearly one. The initialization and learning is done using all the labeled data (for each class) and known classes are learned independently of one another. The unlabeled data points are used to initialize and learn components which are non-predefined and are **not** associated with any known class as they are initially associated with the unknown class. Then, all components are "merged" into a final model encompassing all of the data for joint EM learning. During this semi-supervised learning process, all components will learn their component natures, i.e. the $\{v_k\}$'s, having the chance for them to switch states and become probabilistically associated with a known class or an unknown class [10]

Complete experimental results, then, would compile the experimental metrics over all possible K-fold cross validation experiments to give an accurate, unbiased, and independent evaluation of the performance of the GMM for these data. However, analysis of such mixtures is difficult even for data sets as small as what we are dealing with here. Therefore we will only present results from a single K-fold cross validation experiment so that reasonable exploration, analysis, and conclusions can be drawn.

3.3 Some Statistical Features of Retinal Images

A goal of this analysis is to determine the relationship between cellular/sub-cellular activity and retinal detachment/reattachment under various treatment and control conditions. Optimally, we would like to obtain features which best capture and characterize the status of the retina and cellular activity (for example, the number of nuclei in a given layer of the retina or the shape and intensity of the muller cells in the internal regions of the retina). For this, edge detectors, blob/shape detectors/counters, as well as a number of advanced image processing methods are being developed which will result in high-dimensional feature vectors for potential use with GMMs. Although our methods are applicable to any general set of features, for now, we will focus attention on a simple and novel statistical characterization of retinal images and resulting statistical summaries.

Consider a retinal image such as the one in Figure 1. Since the image is obtained as a collection of 3 different channels (using 3 different stains) labeled red, green, and blue, we are able to represent each image as an array of 3 images (although for this report we only focus on the red and blue channels). For each channel in each image, we define the following random variables

X : a random column pixel of the image channel
Y : a random row pixel of the image channel

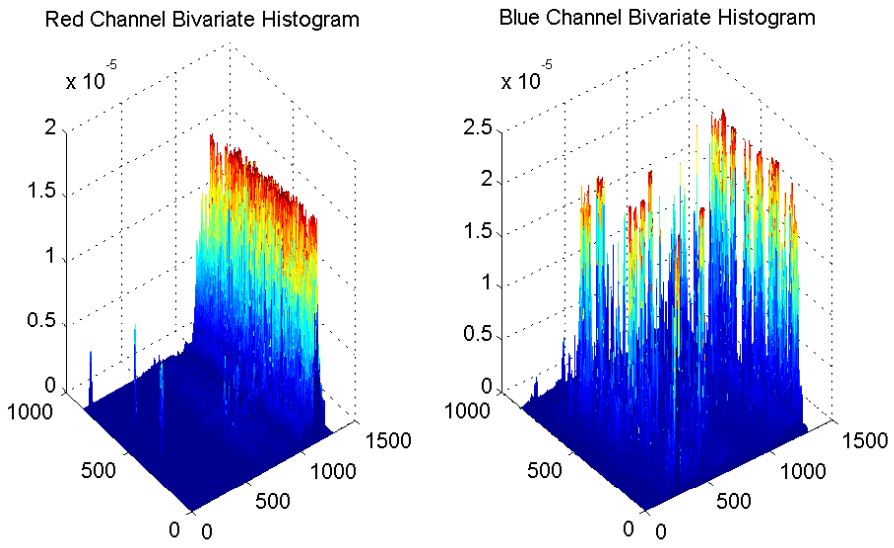

Figure 2: Empirical Histogram of Image for Red/Blue channels

After normalizing each channel so that the sum of the normalized intensities becomes unity, we treat each image channel as a sample bivariate joint density, $P(X,Y)$, observed for the bivariate random variables X, Y as in Figure 2. This representation is essentially an empirical bivariate histogram. Once we have a reasonable probabilistic representation of each image channel, we are able to extract such statistical summaries as: the mean or *center of mass* of the column random variable, X, and the sample covariance of X, Y.

$$\overline{X} = \sum_{i=1}^{n} X_i P(X_i) \tag{15}$$

and

$$\begin{aligned} COV(X,Y) &= \begin{pmatrix} \widehat{\sigma}_X^2 & \widehat{\sigma}_{X,Y} \\ \widehat{\sigma}_{X,Y} & \widehat{\sigma}_Y^2 \end{pmatrix} \\ &= \sum_{i=1}^{n}\sum_{j=1}^{m} P(X_{ij},Y_{ij}) \left(\begin{pmatrix} X_{ij} \\ Y_{ij} \end{pmatrix} - \begin{pmatrix} \overline{X} \\ \overline{Y} \end{pmatrix} \right) \left(\begin{pmatrix} X_{ij} \\ Y_{ij} \end{pmatrix} - \begin{pmatrix} \overline{X} \\ \overline{Y} \end{pmatrix} \right)^T \end{aligned} \tag{16}$$

where n is the number of columns, m is the number of rows and, $P(X_i) = \sum_{j=1}^{m} P(X_i, Y_j)$.

Once we have an empirical representation of each image channel, we are able to extract 2 main statistical summaries of each image channel: the mean or *center of mass* of the column random variable, X, as well as the sample covariance of X, Y.

We now provide a practical interpretation of the statistical features we have developed. Recall that these features are extracted for each of the channels available for each image. Ideally, each channel is staining (and thus illuminating) the cellular components of various layers of the retina. For example, the red channel highlights the Outer Nuclear Layer (ONL) where as the blue channel highlights the Muller cells located in the inner parts of the retina. For reasons still under investigation, there is evidence to suggest that the ONL layer of the retina becomes unstable, falls apart, and diffuses into the inner parts of the retina. Likewise, the Muller cells tend to elongate and spread into the outer parts of the retina. To statistically capture these events, the statistical features we extract measure 2 important occurrences for each channel.

First, the mean column value conveys where the mean value or *center of mass* of each channel is. Presumably, the effects of detachment over extended periods of time are to shift the mean location of the ONL layer (as represented in the red channel) towards the center of the retina. Conversely, the effect of detachment is the opposite for the Muller cells as they elongate and grow towards the outer part of the retina.

Before describing what we hope the covariance features capture, we must note that we don't use the covariance matrix as features explicitly. First, we begin with an eigenvalue/eigenvector decomposition of the 2*2 covariance matrix. Then, we use the ratio of the 2 eigenvalues (specifically the ratio of Y to X) as the so called *Eigen Feature Ratio* (EF). The EF represents the spread of the X or column random variable relative to the Y or row random variable. As the effects of retinal detachment on set, we expect that the EF for the red channel will tend to some value less than or equal to 1 as the ONL layer becomes more spread into the inner parts of the cell. Regenerated retinas will have EF values significantly larger than 1 (usually around 2-5).

3.4 Analysis With Statistical Features

Here, we use the statistical features to analyze classes of retina images. In analyzing the fitted mixture model, we will explore the mixture components to look for similarities, differences, and any biological conclusions that can be drawn from distribution of the labelled data samples over components and classes. Additionally, we will explore misclassified unlabeled samples and how the ownership of the unlabeled samples is distributed between the components and classes as well.

Each of the samples is processed automatically to orient and extract the statistical features. After feature selection which we will not detail here, the feature vector we use is $\underline{x} \doteq (EFR, \overline{x}_R, \overline{x}_B)$ where EFR is the eigenvalue ratio feature for the Red Channel and $\overline{x}_R$ and $\overline{x}_B$ are normalized row averages for the Red and Blue channels respectively.

3.4.1 Inferential Results

After we fit a GMM and do the statistical inference for class prediction, we assess the accuracy of the statistical inference that is produced, in terms of the confusion matrix which is presented in the table below. A confusion matrix details how each unlabeled sample is classified, by class. This evaluation of the model is an indication of how separable the classes are when characterized by a particular feature vector. For this GMM fit, we can see that the *probability of correct classification*, denoted P_{cc}, is the proportion of unlabeled samples which are correctly classified given that they are classified to a known class and truly belong to a known class. Another useful metric is to consider the proportion of unlabeled samples which are classified as unknown given that they are truly known which we denote $P(Unk|Known)$.

	Predicted Class		
Observed Class	3D	3D+28R	Unknown
3D	.75	.25	0
3D+28R	0	.92	.08

For this GMM fit, we find that $P_{cc} = 0.9333$ and $P(Unk|Known) = 0.0625$. The relatively good classification rates provide a measure of confidence in our modeling effort. In actuality, only 2 classification errors were made: (i) a 3D sample was classified as 3D+28R and (ii) a 3D+28R was classified as unknown. Note that these samples are presented in Figure 3 and Figure 4 respectively.

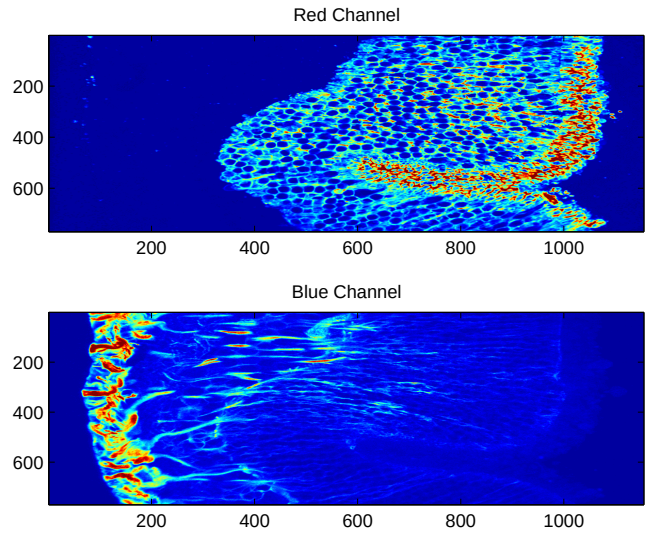

Figure 3: 3D unlabeled sample mis-classified as 3D+28R

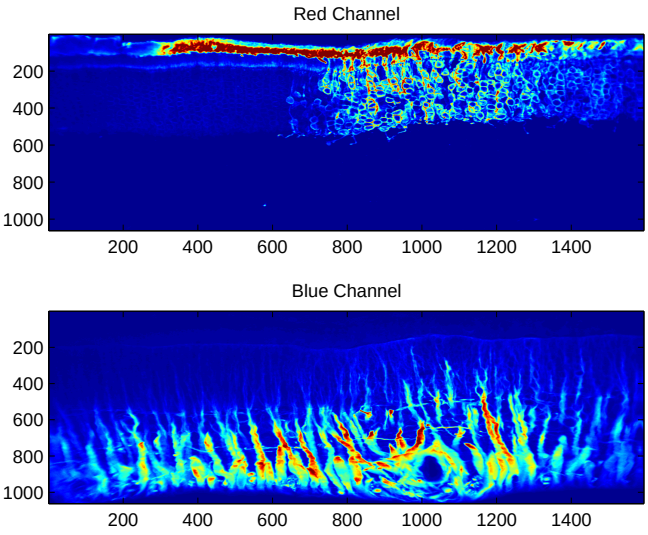

Figure 4: 3D+28R unlabeled sample mis-classified as Unknown

It should be clear that the latter classification error is, perhaps, the most serious since there is not a true *unknown* class in the data. In order for this to occur, a mixture component which is based on unlabeled data and associated with the unknown class would have have to have remained non-predefined throughout the entire EM learning process. To better understand this, we next explore the GMM fit and its components more closely. We will see that this unknown class error is not as serious as we may expect, and in fact, it is a very good aspect of the GMM and EM methods for finding outliers, anomalous data points, and unknown data classes, which this misclassified sample represents (this error is the result of poor image processing, which we discuss in the next section).

3.4.2 GMM Component Analysis

In addition to understanding the inferential errors made in the previous section, closer analysis of the mixture model components can reveal similarities, differences, and practical observations about and between each known class and its local regions. To understand each component, we consider several pieces of information viz. we consider the mean EFR, the component nature, the class each component was initialized on, the final class each component probabilistically owns, and the number of samples the component owns. The GMM fit numerical information is summarized in the below table.

	Mixture Component							
Parameter	1	2	3	4	5	6	7	8
Weight	.0303	.2109	.0840	.2276	.1057	.1687	.0154	.1492
EF Red	.4646	1.520	1.1012	3.4723	.5652	2.2406	.1464	4.7634
Initial Class	1	1	1	2	2	Unk	Unk	Unk
Final Nature	1	1	1	1	1	1	0	1
Final Class	1	1	1	2	2	2	Unk	2
Samples Owned	1	10	3	12	12	12	1	10

3.4.3 Component Visualization

Simply looking at each component's numerical values can be very enlightening and revealing. However, it is also beneficial to *visually* inspect each component by actually looking at the samples owned by each component. The goal in doing so is to determine if there are any patterns, similarities, or differences that we can visually notice. The information may be insightful and not contained in the feature vector information contained in the GMM model components.

This can be achieved in several ways. Obviously, one could look at the entire set of samples owned by the mixture component or a subset of such samples. This has the drawback of looking at, what may amount to, an enormous amount of samples. To combat this, we offer a simple method by which we can visualize each component in a single image. We will consider a *smoothed* image by taking all of the samples owned by a particular component and creating an "average" or smoothed component by (1) interpolating each image so that they are the same size and then (2) giving each image equal weight and taking the average of the images. Note that features are not extracted from interpolated images and that this is only used for visualization purposes.

Mixture Component 1
Mixture component 1 was initialized using data from class 3D. In the end, it ends up only owning 1 sample, it is predefined, and it is associated with class 3D. It seems to represent a local region of

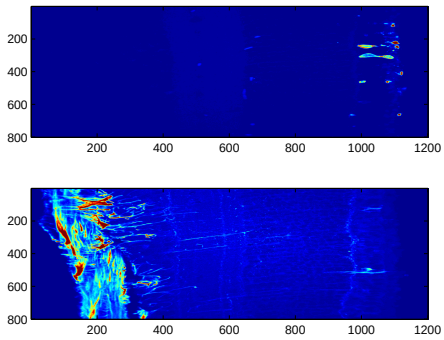

Figure 5: Mixture Component 1 smoothed image-a single sample, class 3D

class 3D which has a completely deteriorated ONL layer. Since the component only owns a single sample, the smoothed image is just a single image itself (Figure 5).

Mixture Components 2,3
Mixture component 2 was initialized using data from class 3D and owns a majority of the samples from class 3D (10 samples: 7 labeled, 3 unlabeled). After all EM learning, this component is predefined and associated with class 3D. It seems to represent a large, local region of class 3D which has a ONL layer which is deteriorating but not completely destroyed as indicated by the EFR (1.52, which is clearly larger than 1). Mixture component 3 is very similar to component 2 which the EFR (1.1012) value confirms This component owns the remaining labeled samples from class 3D (3 samples).

Based on the smoothed images for these components (Figures 6 and 7), it should be clear that neither component represents a unique local region of the 3D class. That is, we would have hoped that each component would have captured images which have similar curvature of the ONL layer (say one component for convex ONL's and another for concave ONL's). Because our current set of feature vectors does not explicitly-or even implicitly-capture the nature of the curvature of the ONL layer but simply the amount of spread, it is not surprising that we do not capture unique local regions with these 2 components. They do, however, capture a local region of class 3D which show significant deterioration of the ONL layer. Since the grouping is based on the amount of spread, we can conclude that there are 2 local regions of class 3D which have differing amounts of spread for deteriorating ONL layers.

Mixture Component 4,6,8
Mixture components 4,6, and 8 (Figures 8, 9, and 14) all own data from class 3D+28R after all EM learning is complete. These components own a significant majority of the 3D+28R data points (34/46 samples) which are both labeled and unlabeled. Although components 6 and 8 were initialized based on unlabeled data and probabilistically associated with the *unknown* class, during the EM learning process they probabilistically own enough labeled samples from class 3D+28R to reverse their component nature to *predefined* and become probabilistically associated with class 3D+28R.

What is interesting about these model components is that they seem to represent a local regions of the 3D+28R class which have had either significant regeneration of the ONL layer *after* retinal re-attachment (component 8 say) or have halted the degeneration of the ONL layer

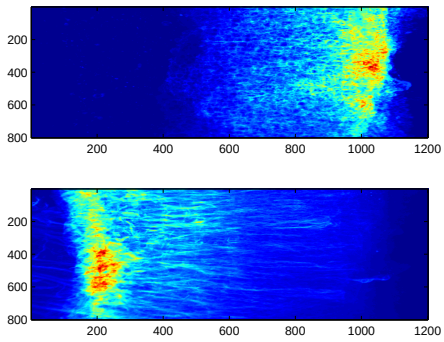

Figure 6: Mixture Component 2 smoothed image, class 3D

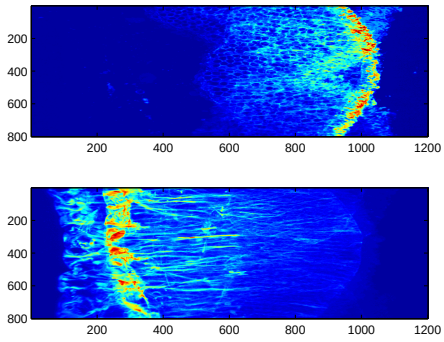

Figure 7: Mixture Component 3 smoothed image, class 3D

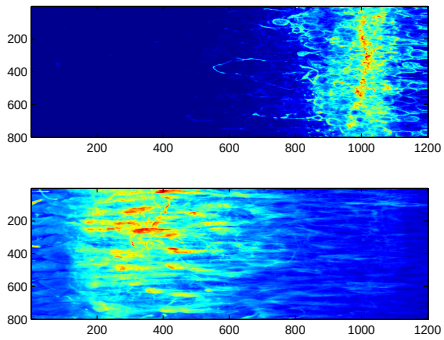

Figure 8: Mixture Component 4 smoothed image, class 3D+28R

(component 4 and 6). Based on the *smoothed* images, components 4 and 6 seem to differ in only a single respect. Component 4 has a much more clearly defined ONL layer as opposed to component 6 which has a much more spread ONL layer. Since the smoothed images don't clearly convey what precisely these components represent, we can always look at all of the samples owned by each of these components for more precise conclusions.

When we inspect all of the images from components 4 and 6 we find some very interesting results that which we would not expect when looking at the smoothed images. That is, component 4 seems to be a more homogenous component representing samples which have had slowed/stopped ONL deterioration as opposed to component 6 which seems to represent a mixture of samples from class 3D+28R. When we consider all of the images, we actually find that component 4 owns samples which are entirely different (Figures 10, 11). Component 4 owns a mixture of samples which have both ONL regeneration and also slowed/stopped ONL deterioration which are entirely different local regions of class 3D+28R. On the other hand, component 6 is far more homogenous which seem almost exclusively represent samples which have stopped/halted ONL deterioration. That is, it only owns a single sample for which there is tacit ONL regeneration (Figure 12). The remaining samples owned by component 6 are all about the same as a sample in Figure 13. Component 4 seems to represent a mixture of 2 local regions of class 3D+28R whereas component 6 almost exclusively represents a local region with only slowed/stopped ONL deterioration. Inspecting only the smoothed images does not reveal this.

Component 8 (Figure 14) looks very similar to component 6 even though EFR values are rather different. The mechanism by which these components are separated is the EM learning process. That is, component 8 is initialized only on unlabeled data samples. Through the EM learning process, it becomes associated with known class 3D+28R, by way of owning labeled data points from this class (as indicated by the final ownership of 10 samples: 7 labeled, 3 unlabeled). This component, unlike components 4 and 6, represents samples which have good regeneration of the ONL layer. Visually inspecting all of the samples owned by component 8 confirms this. Component 8 represents a unique local region of class 3D+28R which has had good ONL regeneration.

Mixture Component 5

Mixture component 5 is an interesting and particularly appealing component which is, from beginning to the end, associated with class 3D+28R. Based on the smoothed image (Figure 15), this component seems to own samples (12 samples: 9 labeled, 3 unlabeled) which have not had any ONL regeneration and, quite possibly, continued ONL degeneration. This would be a local region of class

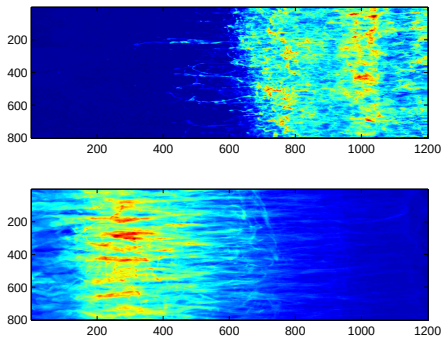

Figure 9: Mixture Component 6 smoothed image, class 3D+28R

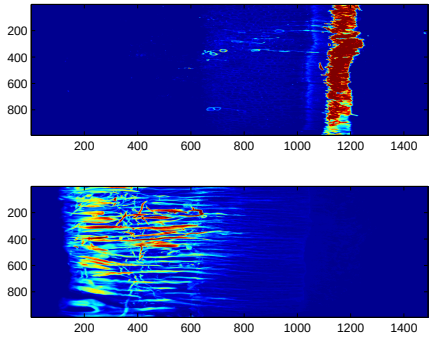

Figure 10: A single sample from Mixture Component 4 , class 3D+28R

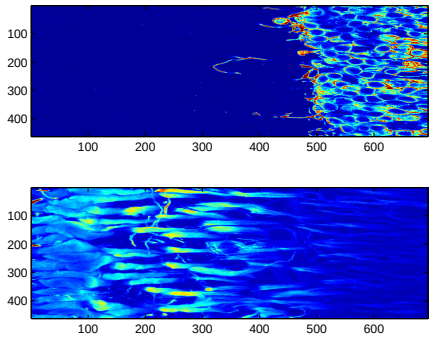

Figure 11: A single sample from Mixture Component 4, class 3D+28R

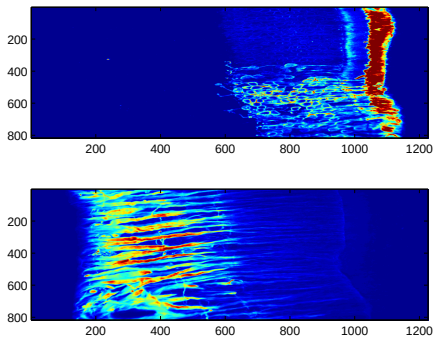

Figure 12: A single sample from Mixture Component 6, class 3D+28R

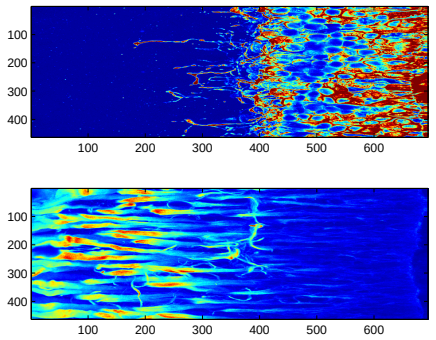

Figure 13: A single sample from Mixture Component 6, class 3D+28R

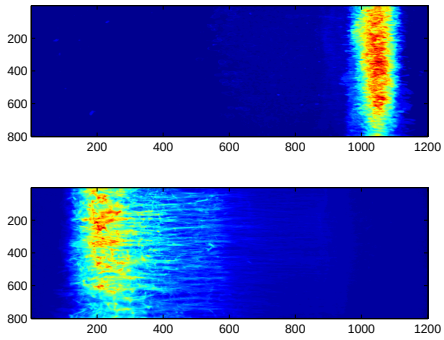

Figure 14: Mixture Component 8 smoothed image, class 3D+28R

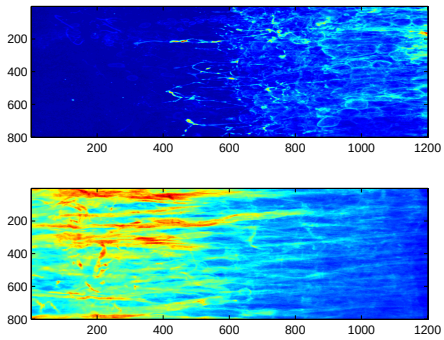

Figure 15: Mixture Component 5 smoothed image, class 3D+28R

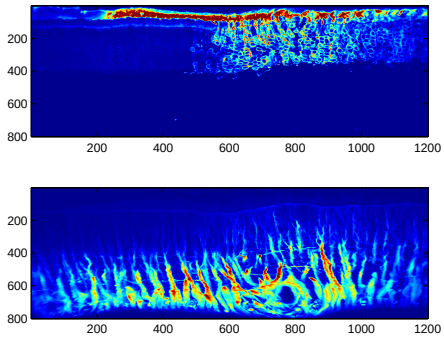

Figure 16: Mixture Component 7 smoothed image-a single sample, class Unknown

3D+28R which seems to resemble class 3D most closely (for which we latter explore analytically). When we visually inspect all of the images owned by component 5, we find that it does represent a mixture of samples which have had continued ONL degeneration and/or slowed/stopped ONL degeneration.

Mixture Component 7

Mixture component 7 (Figure 16) is, again, an encouraging component in the sense that the EM learning method is behaving in a very reasonable manner. It is initialized based on unlabeled data and, as such, it is initially non-predefined. After all EM learning, it is still non-predefined and, as such, associated with some *unknown class*. Note that this is the sample which is mis-classified as unknown. It should be clear by the orientation of this image that there was a major flaw/error in the automated image processing and feature extraction algorithm. That is, the automated image processing failed to orient the image consistently with all of the other image orientations. The GMM model we fit deems this local region of class 3D+28R as unknown as one should hope. The GMM and EM method have determined that the specific sample could not have originated from any known class (as an artifact of poor image processing), is it's own component, and it is degenerately (with probability 1) associated with the unknown class.

3.4.4 Component Testing

We would, a priori, assume that the components found using the GMM and EM learning methods would not be statistically similar to one another. However, we can formally test for component similarity/difference using a multivariate T-test which tests for similarities in the population mean vectors based on sample sizes, mean vectors, and covariance matrices [**?**].

As we suspect from our analysis of the mixture components, we wish to test that Component 5 is equal to the components which represent class 3D by testing equality of Component 5 to Components 1, 2, and 3 simultaneously. We conduct our tests and find that the P-Values for the three tests are .36, 1.5e-007, and .04 for testing the equality of Component 5 to Components 1, 2, and 3 respectively. At any reasonable level of significance (and hypothetically even with proper adjustments to account for the multiple tests being conducted) we would certainly fail to reject the hypothesis of equality for at least Components 5 vs. 1 and possibly even Components 5 vs. 3.

If were to visually inspect the samples from Components 1,2,3, and 5, we would see that the conclusion that Component 1 and Component 5 are not significantly different is quite reasonable since both components represent images that have a highly (if not entirely degenerated ONL) ONL layers found expressed in the red channel of each image. The fact there is strong evidence to reject equality of Component 5 verses Components 2 and 3 is equally encouraging as Components 2 and 3 still have high levels of expression in the red channel indicating ONL layers which are certainly more organized and stable than we see in Component 5 images. These evidence confirms our suspicion that Component 5 is more closely related to class 3D (as represented by Component 1 namely) than class 3D+28R (note that formal statistical tests reveal that there is no evidence to suggest equality of Component 5 to other class 3D+28R Components 4,6, and 8).

3.5 Analysis with Texture Features

Texture features via Gabor Filter analysis is a newly emerging method to develop feature vectors which capture the texture of objects within images should meaningful textures exist [8]. For retinal images, we are looking for textures which capture the curvature, shapes, and contours of various cells in various layers of the retina.

The Gabor Filter Histogram features we use arise through several steps of image processing. First, all image processing is done for each image channel (ie for each R, G, and B channel for the retinal images we consider). For each of the pixels, textures are computed which transforms each pixel into a feature vector where each element is the texture of the pixel relative to the surrounding pixels. Then, all of the pixels for every image that we wish to consider are pooled together and clustered with some high number of components [7]. Then, each of the pixels are classified according the mixture component which owns the pixel. The histogram feature, then, is simply the number of pixels which fall into each cluster category.

For this example, we use the *Normal* retina and *7 Days* of detachment classes. The GMM fits 5 mixtures to the first class and 3 mixtures to the second. The data set contains to truly *unknown* data even though we do initialize components based on the unlabeled data. We rest comfortably since the model does not find an *unknown* class as we would hope. The results of the classification are presented in the table below.

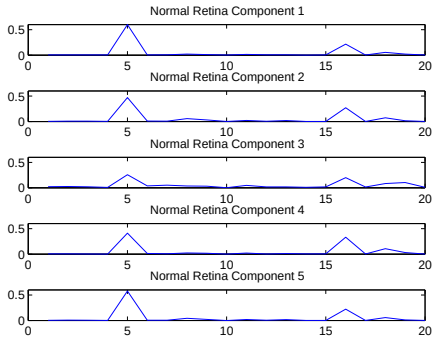

Figure 17: GMM Components for Normal Retinas

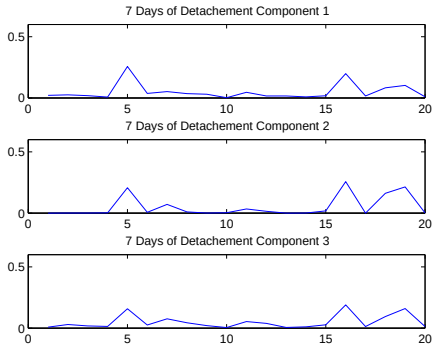

Figure 18: GMM Components for 7 Days of Detachment

	Predicted Class		
Observed Class	Normal	7 Days	Unknown
Normal	.83	.13	0
7 Days	.07	.93	0

Interestingly, we find that there are more *Normal* images are mis-classified as *7 Days*. This would suggest that there are samples from class *7 Days* for which there has been so little deterioration and re-organization that they more closely resemble a normal retina. Inspecting the mixture components reveals something opposite to this conclusion. To inspect the mixture components, we take a different approach than in out previous example. For this, we simply plot the 20 dimensional mean feature vector for each class (on the same scale) to visually look for differences and similarities.

When we inspect the GMM mean feature vectors for each of the classes, what we find is interesting and counter to what we would expect based on the confusion matrix (misclassified) results. Based on the confusion matrix, we would expect to see subsets of class *7 Days* of retinal detachment (at least summarized my mean feature vectors) which resemble *Normal* retinas. What we find, at least based on the labeled training data that is, that there is a sub-region of the *Normal* retinas that more closely resembles the *7 Days* of detachment class. This is indicated by GMM component #3 which has a mean feature vector which is more like the mean feature vectors of the components associated with *7 Days* of retinal detachment. The inference that we extend to the misclassification errors is that there must be unlabeled samples which are *Normal* which more closely resemble the *7 Days* of detachment.

4 Conclusions and Future Research

What is lacking in this report is the biological significance and discussion of these results. That is, what is special about the classes and local regions within any single class that the GMM has found? What is special about the equality relationship between a local region (as represented by Component 5) within the class 3D+28R and a local region (represented by Component 1) within the class 3D? These are questions which the GMM cannot answer with the given feature vectors comprised only of EFR or the Gabor filter histograms. What is required to better answer these questions are features that are biologically significant to better characterize the cellular and sub-cellular behavior within the retina. With such information, biologically significant conclusions beyond what we present here, are possible. This aspect of the image processing segment of the ITR project is crucial to being able to better answer biologically relevant questions.

Along these lines of research, better automated image processing software is desired. Recall that a one of the few classification errors that was made is due to an image sample for which the automated image processing software failed to correctly orient the image, thus extracting erroneous features which make the image not comparable to other samples. As such, the unlabeled data point created it's own mixture component (Component 7) and deemed the sample as unknown. Although this is a promising and delightful result of the GMM and related semi-supervised EM learning method, it is a disaster for the image processing software we use at present. Better features and better image processing (specifically for orientation and image size alignment) are key to answering biologically significant questions.

Another area of potential biological research is to consider the retina status (in addition to length of detachment, re-attachment, and experimental conditions) as either abnormal, normal related to the *vision status* of the retina. That is, given the length of detachment, re-attachment, and experimental treatments, what is the vision status of the retina. Simply put, is it still functioning sufficiently such that the host subject still has some or all vision capabilities.

5 Acknowledgements

We wish to thank the National Science Foundation and Dr. B. S. Manjunath for the financial support as well as the opportunity to work on this project. We also wish to thank Mr. M. Aleem Siddiqi for his work developing the concepts behind the statistical features we use. The Gabor filter historgram features were provided by Dr. Zhiqiang Bi with the theoretical foundations layed primarily by Dr. B. S. Manjunath.

References

[1] M. Aitkin and D.Rubin, "Estimation and Hypothesis Testing In Finite Mixture Models," Journal of the Royal Statistical Society, Series B (Methodological), Vol. 47, No.1, pp 67-75, 1985.

[2] M. Escobar and M. West, "Bayesian Density Estimation and Inference Using Mixtures," Journal of the American Statistical Assocaition, Vol. 90, No. 430, pp 577-588, 1995.

[3] A. Gelman et al, *Bayesian Data Analysis*, Chapman and Hall, Boca Raton, 2001.

[4] P. Green and S. Richardson, "On Bayesian Analysis of Mixtures with than Unknown Number of Components," Journal of the Royal Statistical Society, Series B (Methodological), Vol. 59, No.4, pp 731-792, 1997

[5] T. Hastie et al, *Elements of Statistical Learning*, Springer, New York, 2001.

[6] T. Leaonard and J. Hsu, *Bayesian Methods*, Cambridge Univeristy Press, Cambridge, 2001.

[7] G. McLachlan and T. Krishnan, *The EM Algorithm and Extensions*, John Wiley and Sons, New York, 1997.

[8] B.S. Manjunath et al, "Using Texture To Analyze and Manage Large Collections of Remote Sensed Image and Video," Applied Optics, Vol. 43, No. 2, pp 210-217, 2004.

[9] D. Miller and J. Browning, "A Mixture Model and EM-based Algorithm for Class Discovery, Robust Classification, and Outlier Rejection in Mixed Labeled/Unlabeled Data Sets," IEEE Trans. on Pattern Anal. and Machine Intell., pp 1468-1483, 2003.

[10] D. Miller and S. Frame, "Machine Learning for Robust Automatic Target Recognition: Phase I Final Report," Phase 1 Final Report for U.S. Air Force Research Laboratory Contract FA8650-04-M-1659, 2005.

[11] N. Dean et al, "Using unlabeled dat to update classificatin rules with applications in food authenticity studies," Journal of the Royal Statistical Society, Series C (Applied Statistics), Vol. 55, No. 1, pp 1-14, 1-14.

[12] A. Rencher, *Methods of Multivariate Analylsis*, John Wiley and Sons, New York, 1995.

[13] S. Ross, *Simulation*, Academic Press, San Diego, 1997.

[14] M. Stephens, "Bayesian Analysis of Mixture Models with an Unknown Number of Components-An Alternative to Reversible Jump Methods," The Annals of Statistics, Vol. 28, No. 1, pp 40-74, 2000.

[15] X. Zhu, "Semi-Supervised Learning Literature Survey," Tech. Report No. 1530, Computer Sciences Department, University of Wisconsin, Madison, 2006.