Multi-Instance Mixture Models and Semi-Supervised Learning

James Foulds[*] Padhraic Smyth[*]

Abstract

Multi-instance (MI) learning is a variant of supervised learning where labeled examples consist of bags (i.e. multi-sets) of feature vectors instead of just a single feature vector. Under standard assumptions, MI learning can be understood as a type of semi-supervised learning (SSL). The difference between MI learning and SSL is that positive bag labels provide weak label information for the instances that they contain. MI learning tasks can be approximated as SSL tasks by disregarding this weak label information, allowing the direct application of existing SSL techniques. To give insight into this connection we first introduce multi-instance mixture models (MIMMs), an adaption of mixture model classifiers for multi-instance data. We show how to learn such models using an Expectation-Maximization algorithm in the case where the instance-level class distributions are members of an exponential family. The cost of the semi-supervised approximation to multi-instance learning is explored, both theoretically and empirically, by analyzing the properties of MIMMs relative to semi-supervised mixture models.

1 Introduction

Multi-instance (MI) learning [5] is a variant of supervised learning that has received significant attention in the machine learning literature. While in traditional supervised learning the learning instances are represented as feature vectors, in MI learning the examples are represented as *bags* (i.e. multisets) of feature vectors. Training data consists of labeled bags, and the task is to predict the labels for unseen bags. Labels are not typically provided for the individual feature vectors (referred to as “instances”) though it is usually assumed that the instances have hidden labels that in some way determine the bag labels.

A *multi-instance assumption* is an assumed relationship between instances and bag labels. The most commonly used MI assumption, which Weidmann et al. [17] call the *standard MI assumption*, is that a bag is labeled positive if and only if it contains at least one positive instance. This assumption was used by Dietterich et al. [5] in the context of the “musk” prediction task. The musk task is to predict whether a given molecule will have the desired property of being a musk, i.e. emitting a musky smell. The learning task is difficult because molecules can assume different conformations (shapes) by rotating their internal bonds, and it may be difficult to tell which conformation was responsible for biological activity. Dietterich et al. represent a molecule as a bag of feature vectors, with each feature vector representing a conformation. The standard MI assumption applies because a molecule is active if and only if there exists a conformation that binds to the target binding site.

This paper is concerned with the relationship between MI learning and semi-supervised learning (SSL), the class of learning tasks where both labeled and unlabeled examples are available to the learning algorithm at training time. MI learning under the standard assumption can be viewed as a variant of SSL, since all instances in negative training bags must, by the assumption, be labeled negative, while instances in positive training bags are not directly labeled [24]. MI differs from SSL in that positive bag labels provide weak label information for the instances that they contain, namely that at least one instance in the bag is positive. An implication of this is that MI learning problems can be approximated as SSL problems by disregarding the information contained in positive bag labels.

We explore this connection from the perspective of generative probabilistic models of multi-instance data. The *generative* approach to classification involves modeling the joint distribution of the input domain and the output domain $P(\mathbf{x}, y)$, as opposed to *discriminative* learning which involves either modeling the posterior class probabilities $P(y|\mathbf{x})$ or directly learning a decision boundary to separate the classes. Generative models are so named because it is possible to sample from them to create synthetic data. Although generative classifiers are typically less accurate at classification than discriminative classifiers when large amounts of labeled data are available, they can do better when small amounts of data are available [12], and have the advantage of being able to make use of unlabeled data and handle missing attribute values.

In this paper, we introduce a unified framework for a class of generative models that respect the standard MI assumption. These models adapt the well-known mixture model classifier to handle MI data, and will be referred to as multi-instance mixture models (MIMMs). We show how to learn these models via an EM algorithm in the case where the instance-level classes are expectation-parametrized exponential family distributions. Using this framework, we investigate the cost of the SSL approximation to MI learning both theoretically and empirically.

In Section 2 we give some background on MI learning. Sections 3 and 4 introduce multi-instance mixture models

[*] Department of Computer Science, University of California, Irvine. {jfoulds,smyth}@ics.uci.edu

and describe an EM learning algorithm for them. Sections 5 and 6 provide some theoretical insight into MIMMs, showing the connection to the well-known diverse density model for MI learning [11] and proving a bound on the Bayes error rate of the classifier, respectively. In Section 7, we make use of the MIMM framework to quantify the cost of approximating MI learning problems by SSL problems. We present experimental results in Section 8 and conclude in Section 9. A derivation of the EM algorithm is given in the Appendix.

2 Background

Multi-instance learning was originally formulated by Dietterich et al. [5] for the aforementioned musk prediction task. Subsequently, MI learning has been applied to diverse application areas including object detection [15], text classification [21], and contextual advertising [23].

Connections can be made between MI learning and other supervised learning frameworks. For example, MI learning degrades to traditional "attribute-value" learning when the bags are all of size one. De Raedt [3] notes that in terms of generality, multi-instance learning sits between attribute-value learning and fully relational learning. The connection between MI learning and semi-supervised learning, previously described in Section 1, is the subject of this paper.

A large number of algorithms for MI learning have been proposed in the literature. A review of models using MI assumptions other than the standard assumption is given by Foulds and Frank [6]. A common approach is to upgrade propositional algorithms to handle MI data [1, 20, 15, 24]. Maron and Lozano-Pérez [11] proposed diverse density, a discriminative probabilistic framework for MI learning. Faster training procedures for diverse density classifiers were proposed by Zhang and Goldman [22], and Foulds and Frank [7].

Generative models for MI have previously been proposed in the literature. Maron [10] proposed generative models for the musk problem in his PhD thesis, but did not develop learning algorithms for them. Kriegel et al. [9] used a generative model to cluster MI data. In their model, examples are clustered at the bag level. Each cluster consists of an instance-level mixture model that instances are drawn from in an independent and identically distributed (*i.i.d.*) fashion.Yang et al. [21] recently proposed Dirichlet-Bernoulli Alignment (DBA), a generative model for multi-class multi-label multi-instance classification inspired by work in the topic modeling community. The approaches of Kriegel et al. and Yang et al. are closely related to the present work in that instances in each bag are assumed to be generated i.i.d. from instance-level mixture models, and the model parameters are learned via an EM algorithm. The key difference is that in the models that we consider, bags are assumed to be labeled according to the standard MI assumption. Yang et al. describe an approximate variational EM algorithm for their model in the case where the underlying mixture model is a mixture of multinomials. In this paper we present a tractable exact EM algorithm that applies whenever the instance-level mixture components belong to an expectation-parametrized exponential family.

3 Multi-Instance Mixture Models

In this section we introduce a simple but intuitive framework for a class of generative MI models. In these models, the instances in each bag are generated i.i.d. from a mixture distribution, the components of which correspond to instance-level classes. Bags are then labeled via the standard MI assumption. The bag sizes are assumed to be fixed. These models, which we refer to as multi-instance mixture models (MIMMs), can be understood to be an adaption of semi-supervised mixture model classifiers to the MI learning scenario, analogous to Zhou and Xu's adaption of a semi-supervised SVM to handle MI data (MissSVM) [24]. We show how to learn the parameters of MIMMs via an EM algorithm in the case where the instance-level class distributions belong to an expectation-parametrized exponential family.

MIMMs are naive in the sense that they assume that instances are conditionally independent of each other given their class labels (note that *attributes* need not be conditionally independent). The conditional independence assumption allows us to avoid summing over the exponential number of possible labels in a positive bag in the E-step of the EM algorithm. This assumption is of course not always justified in practice. For example, in the musk domain we would expect the instances in a bag to be dependent, as they correspond to conformations of the same molecule. However, naive classifiers are often useful to explore and can still often perform well even when their independence assumptions are invalid, as evidenced by the practical success of the naive Bayes classifier.

More formally, let N_B be the number of bags to generate, and S_i $(i = 1 \ldots N_B)$ be the number of instances in bag i. We assume that N_B and the S_i's are fixed and known ahead of time, although it would be straightforward to introduce a distribution over the S_is. Let $P(\mathbf{x}|z = 1; \theta)$, $P(\mathbf{x}|z = 0; \theta)$ be the distributions for instances of the positive and negative classes, respectively. Here, $\mathbf{x}$ refers to an instance, and $z \in \{0, 1\}$ refers to an instance label. We will denote the jth instance of the ith bag as $\mathbf{x}_{ij}$, its label as z_{ij}, and the label of bag B_i as $y_i \in \{\oplus, \ominus\}$. Capital $\mathcal{X}_i$ and $\mathcal{Z}_i$ refer to the collection of $\mathbf{x}$'s and z's in a bag respectively, with $\mathcal{X}_i = \{\mathbf{x}_{i1}, \ldots, \mathbf{x}_{iS_i}\}$ and $\mathcal{Z}_i = \{z_{i1}, \ldots, z_{iS_i}\}$. For simplicity, let $\pi_p = P(z = 1)$ be the marginal probability of a positive instance, and replace $P(.; \theta)$ with $P_\theta(.)$.

MIMMs assume that the data is generated via the fol-

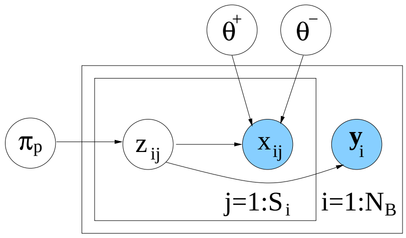

Figure 1: Directed graph of the generative model for MIMMs. Shaded nodes are observed.

lowing process. Given model parameters $\theta = (\theta^+, \theta^-, \pi_p)$, where $\theta^{+/-}$ are the class parameter vectors, generate each bag $B_i = (\mathcal{X}_i, y_i, \mathcal{Z}_i)$ independently via:

1. For each of the S_i instances in B_i (indexed by j):

   (a) Choose a class label $z_{ij} \sim Bernoulli(\pi_p)$

   (b) Generate $\mathbf{x}_{ij} \sim P_\theta(\mathbf{x}|z_{ij})$

2. Label bag B_i via $y_i = \begin{cases} \oplus, \exists z_{ij} \in \mathcal{Z}_i : z_{ij} = 1 \\ \ominus, \text{ otherwise.} \end{cases}$

In other words, for each instance j of bag i flip a weighted coin (a Bernoulli trial with parameter π_p) to decide the label z_{ij}. Then generate $\mathbf{x}_{ij}$ according to the distribution associated with that class label. Bags are labeled according to the standard MI assumption, i.e. positive iff they contain a positive instance. Figure 1 illustrates the generative process with a directed graphical model. Any family of probability distributions defined over the instance space can be used for the instance-level classes. In this work we consider distributions belonging to the exponential family.

By applying Bayes' rule and simplifying, we obtain a classifier with the posterior bag-level class probabilities

$$P_\theta(\ominus|B_i) = \prod_j^{S_i} P_\theta(z_{ij} = 0|\mathbf{x}_{ij}) \,,$$

where $P_\theta(z_{ij} = 0|\mathbf{x}_{ij})$ can be calculated by Bayes' rule, and of course $P_\theta(\oplus|\mathcal{X}_i) = 1 - P_\theta(\ominus|\mathcal{X}_i)$. It is also worth noting that the marginal bag-level class probabilities are a function both of the marginal instance-level class probabilities and the number of instances S, i.e. $P_\theta(\ominus) = (1-\pi_p)^S$, and as $S \to \infty$, $P_\theta(\ominus) \to 0$.

4 EM Learning Algorithm

Given a dataset $D = \{B_1, \dots, B_{N_B}\}$ with labeled bags but unlabeled instances, we would like to find the maximum likelihood estimate[1] of the model parameters $\theta = (\theta^+, \theta^-, \pi_p)$, $\hat{\theta}_{ML} = \arg\max_\theta P_\theta(D)$. First, it is worth noting that since the instances are all generated i.i.d. from the mixture distribution $P_\theta(\mathbf{x})$, if the bag labels are disregarded (and we have no instance-level labels) the likelihood becomes the likelihood for a mixture model, i.e. $P_\theta(\mathcal{X}_1, \dots, \mathcal{X}_{N_B})$ equals

$$\prod_{i=1}^{N_B}\prod_{j=1}^{S_i}\Big(\pi_p P_\theta(\mathbf{x}_{ij}|z_{ij}=1) + (1-\pi_p)P_\theta(\mathbf{x}_{ij}|z_{ij}=0)\Big).$$

However, since the observed data includes the bag labels y_i, the likelihood function becomes more complicated. Since bags are generated independently, the likelihood is a product over the likelihoods of the training bags:

$$L(\theta; D = \{B_1, \dots, B_{N_B}\}) = P_\theta(D) = \prod_{i=1}^{N_B} P_\theta(\mathcal{X}_i, y_i) \,,$$

$$\text{where } P_\theta(\mathcal{X}_i, y_i = \oplus) = P_\theta(\mathcal{X}_i) - P_\theta(\mathcal{X}_i, y_i = \ominus) \,,$$

$$P_\theta(\mathcal{X}_i, y_i = \ominus) = (1-\pi_p)^{S_i} \prod_{j=1}^{S_i} P_\theta(\mathbf{x}_{ij}|z_{ij} = 0) \,.$$

Maximizing the likelihood is equivalent to maximizing the log-likelihood, so $\hat{\theta}_{ML} = \arg\max_\theta \sum_{i=1}^{N_B} \ln P_\theta(\mathcal{X}_i, y_i)$. As with standard mixture models, due to the summation terms inside the logarithm for the terms relating to positive bags we cannot maximize the log-likelihood in closed form. However, the problem lends itself to an application of the Expectation-Maximization (EM) algorithm [4], an iterative optimization technique that is useful for finding maximum likelihood solutions in problems with missing data. Each iteration improves a lower bound on the log-likelihood by alternating between an "E-step" and an "M-step." The E-step finds the function $Q(\theta; \theta^k)$, the expected value of the complete-data log-likelihood with respect to the missing/hidden variables conditioned on both the observed variables and a current estimate of the parameters θ^k, while the M-step maximizes $Q(\theta; \theta^k)$ with respect to θ.

In the case of MIMMs, for bag B_i with instance labels $\mathcal{Z}_i$, the complete-data log-likelihood, which we will denote $L_c(\theta; \mathcal{X}_i, y_i, \mathcal{Z}_i)$, equals $\sum_{j=1}^{S_i}\big(\ln P_\theta(z_{ij}) + \ln P_\theta(\mathbf{x}_{ij}|z_{ij})\big)$ for any assignment of the $\mathcal{Z}_i$s that is consistent with y_i.

With a slight abuse of notation, we write $Z_i = 0$ as shorthand for $\forall z_{ij} \in \mathcal{Z}_i : z_{ij} = 0$. Let $\alpha_i = P_{\theta^k}(Z_i =$

[1] The extension to maximum a posterior estimation is straightfoward and not discussed here.

$0|\mathcal{X}_i)$. Let N^+ be the number of positive *bags* and N^- be the number of *instances* in negative bags. In the following, when index i ranges over 1 to N^+ the index refers to the collection of positive bags, and when i ranges over 1 to N^- the index refers to the collection of instances from negative bags, denoted $\mathbf{x}_i^-$. We now describe the EM algorithm for MIMMs. A derivation of the algorithm is provided in Appendix A.

4.1 E-Step In the following, $E[X]$ refers to the expectation of random variable X. The function $Q(\theta;\theta^k)$, the expected log-likelihood over all bags, can be shown to be

$$\begin{aligned}(4.1)\qquad &\sum_{i=1}^{N^+} \frac{E[L_c(\theta;\mathcal{X}_i,\mathcal{Z}_i)] - \alpha_i L_c(\theta;\mathcal{X}_i,\mathcal{Z}_i=0)}{1-\alpha_i} \\ &+ \sum_{i=1}^{N^-} \ln P_\theta(\mathbf{x}_i^-|z_i=0) + N^- \ln(1-\pi_p) \;.\end{aligned}$$

Here, $E[L_c(\theta;\mathcal{X}_i,\mathcal{Z}_i)|\mathcal{X}_i,\theta^k]$ is the expected complete data log-likelihood that is computed in the E-step of EM for ordinary mixture models,

$$\begin{aligned}(4.2)\qquad &\sum_{j=1}^{S_i} \Big(\gamma_{ij}(\ln \pi_p + \ln P_\theta(\mathbf{x}_{ij}|z_{ij}=1)) \\ &+ (1-\gamma_{ij})(\ln(1-\pi_p) + \ln P_\theta(\mathbf{x}_{ij}|z_{ij}=0)) \Big) \;,\end{aligned}$$

where $\gamma_{ij} = P_{\theta^k}(z_{ij}=1|\mathbf{x}_{ij})$ is

$$\frac{\pi_p^k P_{\theta^k}(\mathbf{x}_{ij}|z_{ij}=1)}{\pi_p^k P_{\theta^k}(\mathbf{x}_{ij}|z_{ij}=1) + (1-\pi_p^k) P_{\theta^k}(\mathbf{x}_{ij}|z_{ij}=0)},$$

the responsibility for instance x_{ij}, i.e. the probability that instance x_{ij} belongs to the positive class.

4.2 M-Step Let $\omega_{ij} = \frac{\gamma_{ij}}{1-\alpha_i}$ for instance j of positive bag i. As we shall see, the ω_{ij}s can be interpreted as responsibilities that have been adjusted to take into account the standard MI assumption. Irrespective of the form of the mixture components, the M-step update for π_p is:

$$\pi_p = \frac{\sum_{i=1}^{N^+}\sum_{j=1}^{S_i}\omega_{ij}}{\sum_{i=1}^{N^+} S + N^-} \;.$$

We now give the M-step updates for MIMMs where the density for each class belongs to an expectation-parametrized exponential family. This EM algorithm is closely related to the EM algorithm for mixtures of exponential family distributions [13]. A set of probability distributions is an exponential family if its densities can be written in the form $p(\mathbf{x};\theta) = a(\theta)^{-1} b(\mathbf{x}) e^{\theta^\intercal t(\mathbf{x})}$. Under certain conditions on its natural parameter space (convexity, openness) and its sufficient statistics (linear independence), an exponential family distribution can be written in the "expectation" parametrization $p(\mathbf{x};\phi) = a(\theta(\phi))^{-1} b(\mathbf{x}) e^{\theta(\phi)^\intercal t(\mathbf{x})}$, where the mapping $\theta \to \phi = E[t(\mathbf{x})|\theta]$. Let each instance-level class $p^c(\mathbf{x}|\phi^c)$, $c \in \{+,-\}$, be such a distribution. The M-step updates are:

$$\begin{aligned}\phi^+ &= \frac{\sum_{i=1}^{N^+}\sum_{j=1}^{S_i}\omega_{ij} t^+(\mathbf{x}_{ij})}{\sum_{i=1}^{N^+}\sum_{j=1}^{S_i}\omega_{ij}} \\ \phi^- &= \frac{\sum_{i=1}^{N^+}\sum_{j=1}^{S_i}(1-\omega_{ij}) t^-(\mathbf{x}_{ij}) + \sum_{j=1}^{N^-} t^-(\mathbf{x}_j^-)}{\sum_{i=1}^{N^+}\sum_{j=1}^{S_i}(1-\omega_{ij}) + N^-} \;.\end{aligned}$$

As examples, we provide the M-step updates for several useful special cases of the exponential family, namely discrete naive Bayes, multivariate Gaussian and first order Markov models. These examples show how MIMMs can easily be applied to MI learning scenarios with different types of instance-level data such binary, continuous or sequential data.

Naive Bayes A simple but useful example is the naive Bayes classifier for discrete data. For clarity we consider the case where, conditioned on the class $c \in \{+,-\}$, each of the d attributes is a conditionally independent Bernoulli variable parametrized by λ_k^c, although the result is similar with multinomial attributes. For each class, we have $P_{\lambda^c}(\mathbf{x}) = \prod_{k=1}^d (\lambda_k^c)^{\mathbf{x}_k}(1-\lambda_k^c)^{1-\mathbf{x}_k}$. In exponential family form, $P_{\lambda^c}(\mathbf{x}) = \big(\prod_{k=1}^d (1-\lambda_k^c)\big)\exp\big(\sum_{k=1}^d \mathbf{x}_k \ln(\frac{\lambda_k^c}{1-\lambda_k^c})\big)$, so the natural parameter is $\theta^c = [\ln(\frac{\lambda_1^c}{1-\lambda_1^c}),\ldots,\ln(\frac{\lambda_d^c}{1-\lambda_d^c})]^T$, $t(\mathbf{x}) = \mathbf{x}$, and we can recover $\lambda_k^c = \sigma(\theta_k^c) = \frac{1}{1+\exp(-\theta_k^c)}$. We can reparametrize in terms of the expectation parameter $\phi^c = E[t(\mathbf{x})|\theta^c] = E[\mathbf{x}|\lambda_k^c = \sigma(\theta_k^c)\forall k] = [\sigma(\theta_1^c),\ldots,\sigma(\theta_d^c)]^T$. Then the M-step updates are

$$\begin{aligned}\lambda^+ =& [\sigma(\theta_1^+),\ldots,\sigma(\theta_d^+)]^T = \frac{\sum_{i=1}^{N^+}\sum_{j=1}^{S_i}\omega_{ij}\mathbf{x}_{ij}}{\sum_{i=1}^{N^+}\sum_{j=1}^{S_i}\omega_{ij}} \\ \lambda^- =& [\sigma(\theta_1^-),\ldots,\sigma(\theta_d^-)]^T \\ =& \frac{\sum_{i=1}^{N^+}\sum_{j=1}^{S_i}(1-\omega_{ij})\mathbf{x}_{ij} + \sum_{i=1}^{N^-}\mathbf{x}_i^-}{\sum_{i=1}^{N^+}\sum_{j=1}^{S_i}(1-\omega_{ij}) + N^-} \;.\end{aligned}$$

Multivariate Gaussian Consider the case where the mixture components are multivariate Gaussians $P_{\theta^c}(\mathbf{x}_{ij}) = \mathcal{N}(\mathbf{x}_{ij}; \mu^c, \Sigma^c)$. Then the M-step updates are

$$\mu^+ = \frac{\sum_{i=1}^{N^+} \sum_{j=1}^{S_i} \omega_{ij} \mathbf{x}_{ij}}{\sum_{i=1}^{N^+} \sum_{j=1}^{S_i} \omega_{ij}}$$

$$\mu^- = \frac{\sum_{i=1}^{N^+} \sum_{j=1}^{S_i} (1 - \omega_{ij}) \mathbf{x}_{ij} + \sum_{i=1}^{N^-} \mathbf{x}_i^-}{\sum_{i=1}^{N^+} \sum_{j=1}^{S_i} (1 - \omega_{ij}) + N^-}$$

$$\Sigma^+ = \frac{\sum_{i=1}^{N^+} \sum_{j=1}^{S_i} \omega_{ij} (\mathbf{x}_{ij} - \mu^+)(\mathbf{x}_{ij} - \mu^+)^\intercal}{\sum_{i=1}^{N^+} \sum_{j=1}^{S_i} \omega_{ij}}$$

$$\Sigma^- = \frac{\sum_{i=1}^{N^+} \sum_{j=1}^{S_i} (1 - \omega_{ij}) (\mathbf{x}_{ij} - \mu^-)(\mathbf{x}_{ij} - \mu^-)^\intercal}{\sum_{i=1}^{N^+} \sum_{j=1}^{S_i} (1 - \omega_{ij}) + N^-} + \frac{\sum_{i=1}^{N^-} (\mathbf{x}_i^- - \mu^-)(\mathbf{x}_i^- - \mu^-)^T}{\sum_{i=1}^{N^+} \sum_{j=1}^{S_i} (1 - \omega_{ij}) + N^-} .$$

First-Order Markov Model Suppose the instances for each class c are sequences drawn from a first-order Markov model, i.e.

$$P_{\theta^c}(x_{ij}) = P_{\theta_I^c}(x_{ij,1}) \prod_{l=2}^{L_i} P_{\theta_T^c}(x_{ij,l} | x_{ij,l-1}) ,$$

where θ_I^c and θ_T^c are the initial state probabilities and transition probabilities for class c, respectively, and L_i is the length of sequence i. Here, $\delta(s_1, s_2) = [x_1 = s_2]$ is the Kronecker delta function, and $n_{s_1,s_2}(x)$ is the count of transitions from state s_1 to s_2 in sequence x. Then for all states $s, s_1, s_2 \in S$ the M-step update equations are

$$\theta_I^+(s) = \frac{\sum_{i=1}^{N^+} \sum_{j=1}^{S_i} \omega_{ij} \delta(s, x_{ij,1})}{\sum_{s' \in S} \sum_{i=1}^{N^+} \sum_{j=1}^{S_i} \omega_{ij} \delta(s', x_{ij,1})}$$

$$\theta_I^-(s) =$$

$$\frac{\sum_{i=1}^{N^+} \sum_{j=1}^{S_i} (1 - \omega_{ij}) \delta(s, x_{ij,1}) + \sum_{i=1}^{N^-} \delta(s, x_{i,1}^-)}{\sum_{s' \in S} \big(\sum_{i=1}^{N^+} \sum_{j=1}^{S_i} (1 - \omega_{ij}) \delta(s', x_{ij,1}) + \sum_{i=1}^{N^-} \delta(s', x_{i,1}^-) \big)}$$

$$\theta_T^+(s_2 | s_1) = \frac{\sum_{i=1}^{N^+} \sum_{j=1}^{S_i} \omega_{ij} n_{s_1,s_2}(x_{ij})}{\sum_{s' \in S} \sum_{i=1}^{N^+} \sum_{j=1}^{S_i} \omega_{ij} n_{s_1,s'}(x_{ij})}$$

$$\frac{\sum_{i=1}^{N^+} \sum_{j=1}^{S_i} (1 - \omega_{ij}) n_{s_1,s_2}(x_{ij}) + \sum_{i=1}^{N^-} n_{s_1,s_2}(x_i^-)}{\sum_{s' \in S} \big(\sum_{i=1}^{N^+} \sum_{j=1}^{S_i} (1 - \omega_{ij}) n_{s_1,s'}(x_{ij}) + \sum_{i=1}^{N^-} n_{s_1,s'}(x_i^-) \big)} .$$

5 Relationship to Diverse Density

Instead of maximizing (in the frequentist case) the unconditional likelihood of the joint distribution of the examples and their labels, another approach is to learn a discriminative version of a generative model by maximizing the conditional likelihood of the class labels given the examples. For example, logistic regression is the discriminative version of the naive Bayes model with Gaussian components. Two classifiers related in this way are called a *generative-discriminative pair* [12]. In the case of MIMMs, maximizing the class-conditional likelihood gives us the objective function

$$\arg\max_\theta \prod_{B_i : y_i = \oplus} \big(1 - P_\theta(\ominus | B_i)\big) \prod_{B_i : y_i = \ominus} P_\theta(\ominus | B_i).$$

Consider also the objective function for the "noisy-or" diverse density model [11]:

$$\arg\max_\theta \prod_{B_i : y_i = \oplus} \big(1 - r(B_i, \theta)\big) \prod_{B_i : y_i = \ominus} r(B_i, \theta) ,$$

where $r(B_i, \theta) = \prod_j^{S_i} \tilde{r}(\mathbf{x}_{ij}, \theta)$. We can interpret $r(B_i, \theta)$ as $P_\theta(\ominus | B_i)$ and $\tilde{r}(\mathbf{x}_{ij}, \theta)$ as $P_\theta(z_{ij} = 0 | \mathbf{x}_{ij})$, since $\tilde{r}(\mathbf{x}_{ij}, \theta) \in [0, 1]$ in [11], (see also [7] and [19]). The diverse density objective function is the same as the class-conditional likelihood for the discriminative version of a MIMM, with the instance-level class probability $P_\theta(z_{ij} = 0 | \mathbf{x}_{ij})$ being the Bayes mixture model classifier. So the diverse density algorithm can be understood as a class-conditional discriminative classifier that forms a generative-discriminative pair with the MIMM that has the same $P_\theta(z_{ij} = 0 | \mathbf{x}_{ij})$. [2]

[2] Note that Maron and Lozano-Perez instead interpret $r(B_i, \theta)$ and $\tilde{r}(\mathbf{x}_{ij}, \theta)$ as posterior probabilities $P(\theta | B_i)$ and $P(\theta | \mathbf{x}_{ij})$ respectively, where maximizing the diverse density corresponds to finding a maximum a-posteriori (MAP) estimate of θ. However, their interpretation of diverse density is not motivated via a likelihood framework.

6 A Bound on the Bayes Error Rate

In this section, we derive an upper bound on the Bayes error rate for MIMMs, in terms of the instance-level false positive rate P_e^+ and false negative rate P_e^- of the Bayes optimal instance level classifier, the probability of a positive instance π_p and bag size S. Note that the instance-level Bayes error rate is $P_e^+\pi_p + P_e^-(1-\pi_p)$.

We compute the error rate for the following sub-optimal classification rule: given the true model parameters, classify instances by hard-assigning them to their most-likely classes (as in K-means), and then classify bags deterministically based on these hard assignments according to the standard MI assumption. The error rate for this decision rule must by definition be no better than the (optimal) Bayes error rate.

Consider first false-positive (FP) type errors. Suppose we are given a random negative bag B^-, which by definition has instance labels $Z = 0$, i.e. all negative. Now the number k of instances that we predict to be positive is binomially distributed $k \sim Bin(P_e^+, S)$. The classifier makes an FP error if it predicts any of the instances in the bag as positive, i.e. if $k > 0$. So the FP error rate $P_e^{\oplus}$ is $1 - Bin(k = 0|P_e^+, S) = 1 - (1 - P_e^+)^S$.

We now consider false-negative (FN) type errors. These occur when a bag with at least one positive instance has all of its instances predicted to be negative. Firstly, the number n_p of positive instances in a random bag (positive or negative) is distributed $n_p \sim Bin(\pi_p, S)$. Now the probability $P_e^{\ominus}$ of misclassifying a positive bag with n_p positive instances is the probability of predicting that all of its instances are negative, i.e. $P_e^{\ominus} = (P_e^-)^{n_p}(1 - P_e^+)^{S-n_p}$.

Now the error rate is $P_e^{\oplus}P(\ominus) + P_e^{\ominus}P(\oplus)$, where $\oplus$ and $\ominus$ are the positive and negative bag labels, respectively. Putting it all together, the error rate for this decision rule is:

$$(1 - (1 - P_e^+)^S).(1 - \pi_p)^s + \\ \sum_{n_p=1}^{S} Bin(n_p|\pi_p, S).(P_e^-)^{n_p}.(1 - P_e^+)^{(S-n_p)} \; .$$

The error rate for the optimal classifier must by definition be no worse than the above error rate that is achieved by this suboptimal decision rule.

7 The Cost of Approximating MI Learning as a Semi-Supervised Learning Task

As discussed earlier, MI learning can be understood as a variant of semi-supervised learning (SSL) [24]. Under this interpretation, MI training bags contain semi-supervised instance-level label information, i.e. all instances in negative bags are negative, and instances in positive bags are unlabeled. In addition to this, positive bag labels give us further label information, namely that at least one instance in each positive bag is positive. This implies that MI learning problems can easily be transformed into SSL problems by disregarding the instance-level label information given by positive bag labels. A natural question to ask is therefore "what is the cost of treating MI problems as SSL problems?"

Since $P_\theta(X)$ can always be written in mixture form as $P_\theta(X) = \pi_p P_\theta(X|Z = 1) + (1 - \pi_p)P_\theta(X|Z = 0)$, assuming both that the standard MI assumption holds and that the instances were generated i.i.d. is equivalent to saying that the distribution of the data can be represented by a MIMM. There are two straightforward ways to obtain SSL models/algorithms from a MIMM. The first is to treat instances from negative bags as labeled negative instances, and instances from positive bags as unlabeled instances, ignoring the constraint that at least one is positive. The model that results is a semi-supervised mixture model, which we will refer to as an MM ("Mixture Model"). The second method is to learn the mixture density parameters in a completely unsupervised manner. The assignment of instance-level classes to the learned mixture component densities can be achieved by, for example, choosing the assignment that maximizes the mixture model likelihood when the negative instance labels are observed. Since the instances are treated as unlabeled for most of the learning process, we will refer to this as an MMU "(Mixture Model Unlabeled)". We can thus address the above question using the surrogate question "what is the cost of replacing a MIMM with an MM or an MMU?".

As well as discarding potentially useful label information, the MM approximation introduces bias since it ignores the missing data mechanism, which is only valid for likelihood-based methods when the missing data are *missing at random*, i.e. the missing data mechanism does not depend on the missing values [14]. The MMU method does not suffer from this problem. It follows trivially from a result by [2] that MMUs will (almost surely) achieve the Bayes error rate given an infinite number of training bags, assuming that the parametric family of the underlying mixture model is known and is identifiable up to the class assignments. However, MMUs makes less direct use of the label information available, which can increase the variance of their estimates (see Section 8.1).

We now give some intuition on the cost of approximating a MIMM with an MM. Consider the Kullback-Leibler (KL) divergence between a distribution on a discrete random variable W conditioned on the non-occurence of a certain element ('0' here), and the unconditional distribution on W. Assuming that $P(W = 0)$ is non-zero, it is not difficult to show that the KL-divergence is

(7.3)
$$D_{KL}(P(W|W \neq 0)||P(W)) = -\log(1 - P(W = 0)).$$

This applies to a MIMM when $W = \mathcal{Z}_i$, $P(W) = P_\theta(\mathcal{Z}_i|\mathcal{X}_i)$ is the distribution of the label assignments $\mathcal{Z}_i$ in

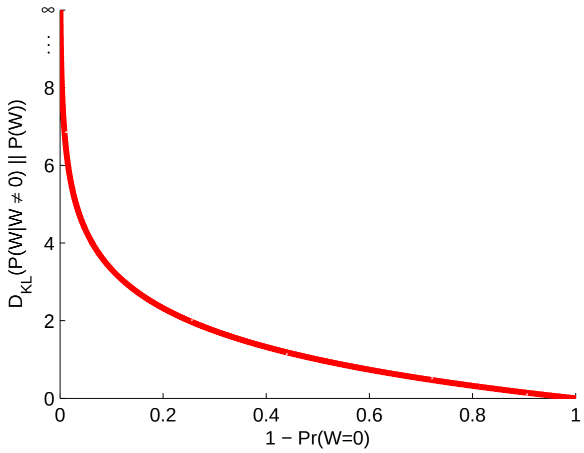

Figure 2: KL divergence from a distribution conditioned on the non-occurrence of a specific value '0' of random variable W to the marginal distribution, versus one minus the probability of that value. More specifically, the information cost (in bits) of ignoring a positive bag label versus the probability that the bag is positive before observing the bag's label.

a multi-instance bag B_i according to the MM, and $W = 0$ is the case where all labels are negative. Then $P(W|W \neq 0)$ is the distribution of the labels given the instances, the model and a positive bag label. Here, the KL divergence quantifies the amount of information that is lost when a positive bag label is ignored.

Interestingly, the KL divergence is dependent only on the probability that the bag is positive, according to the unconditional MM. When the probability of the bag being positive (according to the MM) approaches one, the MM distribution over $\mathcal{Z}$ approaches (in a KL sense) the MIMM distribution over $\mathcal{Z}$. In the other extreme, when the probability of the bag being positive approaches zero the KL divergence approaches infinity. Making use of positive bag labels is therefore important when we have many positive bags that appear likely to be negative given no label information. The relationship between the probability of the excluded value (i.e. the all-negative case) and the KL-divergence is plotted in Figure 2.

Learning a MIMM via the EM algorithm for MMs can be viewed as a variational EM algorithm, where the posterior distribution of the instance labels for positive bags at iteration k, $P_{\theta^k}(\mathcal{Z}_i|\mathcal{X}_i, Z \neq 0)$, is approximated by $P_{\theta^k}(\mathcal{Z}_i|\mathcal{X}_i)$. Equation 7.3, where $P(W) = P_{\theta^k}(\mathcal{Z}_i|\mathcal{X}_i)$, quantifies the cost of the variational approximation in each EM iteration.

From another perspective, we can also get some insight into the relationship between multi-instance learning and semi-supervised learning by simply comparing the likelihoods for MIMMs and MMs. Since MIMMs differ from MMs only in that they are given more observed label information, the likelihood given an observed dataset D for a MIMM must be less than or equal to the MM likelihood. More formally, the likelihoods for MIMMs and MMs can be written as

$$L^{\text{MIMM}}(\theta; D) = (\prod_{i=1}^{N+} (\mathcal{A}_i - \mathcal{B}_i))\mathcal{C}$$
$$L^{\text{MM}}(\theta; D) = (\prod_{i=1}^{N+} \mathcal{A}_i)\mathcal{C}\ ,$$

where $\mathcal{A}_i = \prod_{j=1}^{S_i} \Big(\pi_p P_\theta(\mathbf{x}_{ij}|z_{ij} = 1) + (1 - \pi_p) P_\theta(\mathbf{x}_{ij}|z_{ij} = 0)\Big)$ is the mixture model likelihood for the ith positive bag, $\mathcal{B}_i = (1 - \pi_p)^{S_i} \prod_{j=1}^{S_i} P_\theta(\mathbf{x}_{ij}|z_{ij} = 0)$ is the likelihood of a negative bag with the same x_{ij}s as the positive bag B_i, and $\mathcal{C} = \Big(\prod_{i=1}^{N-} P_\theta(\mathbf{x}_j|z_j = 0)\Big)(1 - \pi_p)^{N-}$ is the mixture model likelihood for the negative instances.

In this form, it is clear that the difference between the likelihood functions for the two models is determined by the $\mathcal{B}_i$s. If we fix the dataset D, in the parts of θ space where the $\mathcal{B}_i$s are very small relative to $\mathcal{A}_i$, such as where π_p is close to one, the MIMM likelihood function approaches the MM likelihood function. Alternatively, if we increase the size of the positive bags, the $\mathcal{B}_i$s will decrease and $L^{\text{MM}}(\theta; D)$ will become a closer approximation to $L^{\text{MIMM}}(\theta; D)$.

8 Experiments

We conducted a set of experiments to *(1)* further investigate the cost of approximating MI learning as an SSL task, and *(2)* to verify the efficacy of the MIMM approach, using both synthetic and real MI data.

8.1 Synthetic Data We generated data from a MIMM using two-dimensional instance-level Gaussian class distributions, with positive and negative means at locations $(0, 0)$ and $(5, 5)$, respectively. Experiments were performed varying the amount of overlap between the classes (diagonal covariance matrices $5I$ and $10I$ for small and large amounts of overlap), π_p (0.2,0.8) and the size of the bags. Note that the prior class probabilities are $P(\ominus|\pi_p = 0.2, S_i = 5) \approx 0.33)$,$P(\ominus|\pi_p = 0.2, S_i = 10) \approx 0.11)$, and $P(\ominus|\pi_p, S_i)$ is effectively zero for $\pi_p = 0.8$, $S_i = 5$ or 10. In each experiment we measured the instance-level error rate versus the number of training bags on a test set of 10,000 instances.

The MIMM, MM and MMU algorithms were trained via EM (ten restarts of up to 200 iterations) to find a maxi-

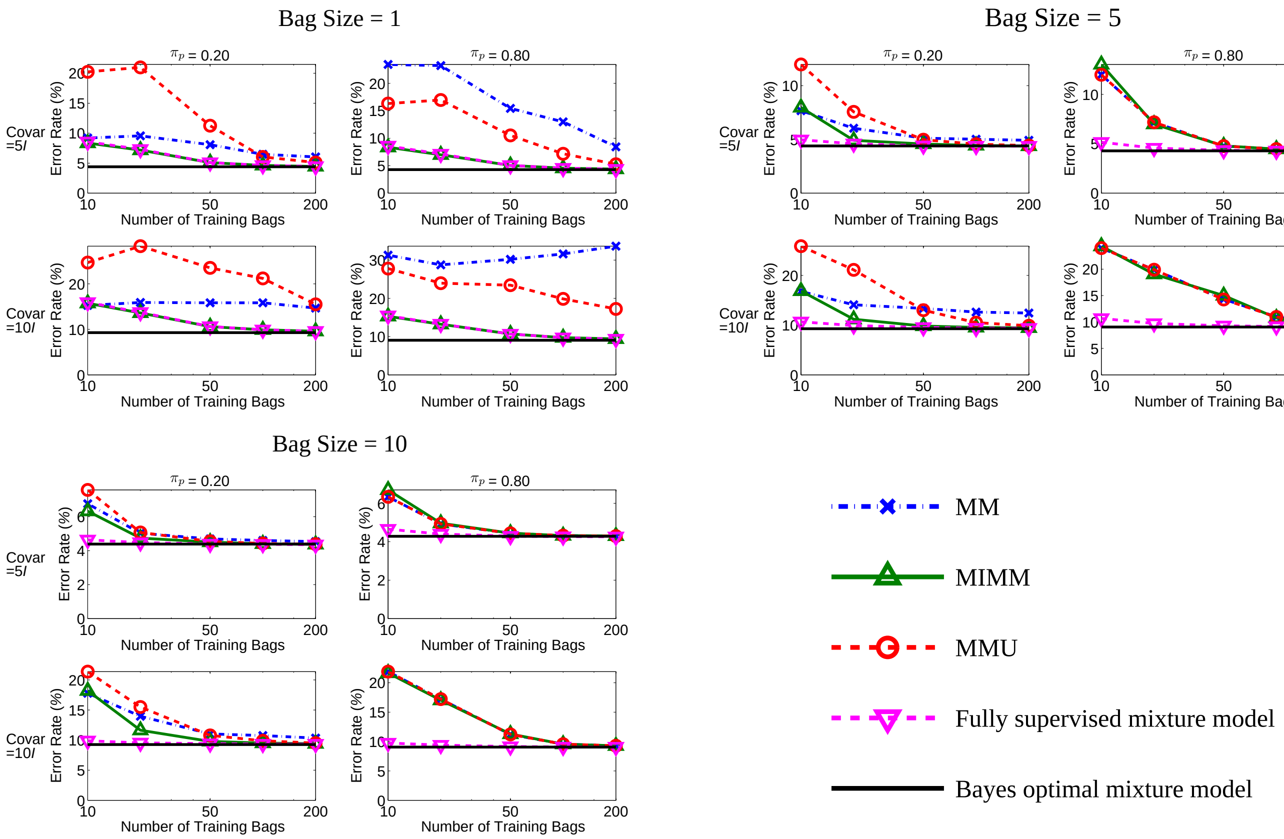

Figure 3: Error rate vs number of training bags for MIMMs and the semi-supervised algorithms on 2D Gaussian data.

mum a-posteriori (MAP) estimate, using the weak conjugate prior on the Gaussian covariance matrices proposed by [8] to avoid singularities, and constraining the covariance matrices to be diagonal. The algorithms were trained on 100 random training datasets for each scenario (the same for each algorithm); in our plots we report the average over these runs. To make learning possible we constrained the datasets to have at least one positive bag, and to have at least one negative bag when the bags were of size one.

The issue of identifiability arises in this setup as there may not be sufficient label information to determine which class label assignment should be applied to the learned class distributions. We remove this relatively uninteresting phenomenon by giving the learning algorithms an "identifiability oracle" that allows them to chose the labeling that gives the best accuracy on the test data. Without the identifiability oracle the results for $\pi_p = 0.2$ were similar to those shown here, while for $\pi_p = 0.8$ the error rate for the MIMM with ten training bags was typically around double the equivalent "oracle" error rate, but still approached the Bayes error by 200 bags, and the error rates for the SSL methods were consistently around 50%.

The results are plotted in Figure 3. The plots show that the improvement of the MIMM over the SSL methods is the greatest with small bags, small π_p and large class overlap. In these cases, the MIMM approaches the Bayes error rate much more rapidly than the MM and the MMU as the amount of training data increases. This is consistent with our theoretical results, as small bags and small mixture parameter values imply a higher prior probability (and consequently higher posterior probability) of negative bags, and large overlap also increases the probability that positive bags "look like" negative bags. With bags of size one, the MIMM reduces to fully-supervised learning. When $\pi_p = 0.2$, the plots show that the MMU overtakes the MM with enough training bags, because more training data means that the MM's bias costs it more relative to the MMU than it gains due to the reduction in variance from access to labels. As we expected from the results of Section 7, the MIMM method approaches the Bayes error rate faster than the semi-supervised methods with small π_p, with the advantage decreasing for large bags and small class overlap. The difference in performance

	MIMM	MM	MMU	EMDD	MILR	MaxDD	MISVM
Musk1	70.90	70.82	53.13	85.4	73.44	86.8	75.94
Musk1 (PCA)	89.08	86.68	56.39	77.82	77.94	71.78	55.92
Musk2	67.63	67.32	47.35	85.6	76.97	85.7	71.78
Musk2 (PCA)	67.32	66.54	59.96	72.95	52.54	68.58	61.73
Elephant	74.40	68.45	52.35	74.80	79.70	80.00	78.85
Elephant (PCA)	69.40	60.30	51.75	76.20	78.15	77.40	50.0
Tiger	61.85	58.45	53.90	72.50	75.60	73.95	81.45
Tiger (PCA)	60.80	59.15	49.60	72.35	73.60	67.20	49.45
Fox	53.20	53.00	48.65	59.65	59.00	60.55	48.60
Fox (PCA)	51.90	52.25	48.55	56.50	53.20	50.70	50.00

Table 1: Accuracy of the MI and semi-supervised algorithms on the musk and image datasets.

between the MIMM and the SSL methods disappears with $\pi_p = 0.8$ for bags of size five and ten. This is a regime where the semi-supervised approximation to multi-instance learning is appropriate, since positive bag labels provide very little extra information and consequently make no difference to classification accuracy.

8.2 Real MI Data We evaluated the Gaussian MIMM, MM and MMU algorithms (with ten restarts of up to 200 iterations) on the two well known musk datasets [5], and on the elephant, fox and tiger image datasets [1]. The musk and image datasets are high dimensional (166 and 229, respectively), which can be problematic for Gaussian mixture model classifiers. To mitigate this, we restricted the Gaussians to have diagonal covariance matrices, and used [8]'s regularization prior on the covariances. We also performed principal component analysis to create versions of the datasets that were reduced to ten dimensions (we allowed full covariance matrices for these datasets). The maxDD [11], EMDD [22], MI logistic regression [20] and MISVM [1] methods were used as baselines, using their default parameters from the WEKA data mining suite [18], except the iterative algorithms (maxDD, EM-DD) were given the same number of restarts and iterations as the MIMM. The algorithms were evaluated using the average of ten repeats of ten-fold cross-validation. Accuracy results are shown in Table 1.

The MIMM performed very well on Musk1 with PCA features, obtaining the highest classification accuracy of the algorithms considered in the experiment. Interestingly, the simple MM method was also very competitive with the dedicated MI algorithms on this dataset. The PCA transformation resulted in reduced accuracy for all of the algorithms for Musk2. Although the MIMM for Musk2 (PCA) had much lower accuracy than the other algorithms that were trained on the untransformed data, it was still competitive on the PCA dataset. The MIMM and the MM performed almost identically on Musk2 – this is most likely due to the larger bags in this dataset. Musk1 has an average of around five instances per bag, while Musk2 has around 65 instances per bag on average. As shown previously in our theoretical results and experiments on synthetic data, the advantage of MIMMs over MMs is reduced for large bags. If instances in a multi-instance dataset are independent as in a MIMM, positive bag labels are of little use in datasets with large bags such as Musk2, unless the mixture parameter is small.

The MISVM's poor performance on the PCA versions of the data can probably be attributed to a lack of parameter tuning. The MIMM and the semi-supervised methods exhibited mediocre performance on the image datasets, relative to the discriminative methods. Generative models are known to be less robust with respect to model misspecification than discriminative methods. The attributes in the image datasets are non-Gaussian, so mixture models with non-Gaussian component densities may be more suitable here.

9 Discussion and Conclusions

In this paper we introduced multi-instance mixture models, a generative framework for MI learning. Learning in this context can be viewed as estimation in the presence of a combination of missing information and constraints, leading naturally to approaches such as EM. We used this framework to investigate the cost of approximating multi-instance learning problems as semi-supervised problems. We showed that the information cost (in bits) of ignoring positive bag labels is minus the logarithm of the probability that the bag was positive before the bag's label was observed. The number of bits of information contained in a positive bag label approaches zero as the probability of the bag already being positive approaches one, and approaches infinity as the probability of the bag already being negative approaches one.

Assuming that both the standard MI assumption holds and that instances are generated i.i.d. is equivalent to assuming that the data distribution can be represented by a MIMM. In this case, disregarding positive bag labels corresponds to approximating the data distribution with a semi-supervised mixture model. The cost of ignoring positive labels is therefore computed with respect to this semi-supervised model. Equation 7.3 implies that similar arguments hold in the non-i.i.d. case. It would be interesting to explore the implications of this in the non-i.i.d. case using alternative generative models. Comparing the likelihood functions for MIMMs and semi-supervised mixture models gives intuition similar to the information theoretic result.

The above results suggest that it can be appropriate to use the semi-supervised approximation to MI learning when the positive bags in the training data tend to be predicted as very likely to be positive by the semi-supervised method. For example, consider a scenario where the positive bags are large, the prior probability of a positive instance π_p is high (and therefore, assuming independence, the prior probability of a positive bag is also high), and there is little overlap between the classes so positive bags are easily recognized by the semi-supervised model. In this case, the positive bag labels are not very informative and the cost of ignoring them is small. On the other hand, if the positive bags are small, positive instances are apriori unlikely and the classes overlap heavily then it may be important to use algorithms that take advantage of multi-instance bag label information. This is supported by our experimental results on synthetic and real datasets.

One implication of this is that there may not be a practical need for MI learning algorithms (based on the standard assumption) that scale up to handle very large positive bags. As the number of instances increases, the probability that at least one of them is positive approaches one, and the amount of information contained in a positive bag label approaches zero bits. Semi-supervised learning algorithms are likely to work just as well as multi-instance learning algorithms on such problems (e.g. see Figure 3).

The experimental results also showed that the proposed MIMM generative models, trained via EM, can be competitive with existing discriminative algorithms, at least in the case where the generative assumptions approximately hold such as on the musk data. The framework is quite general, being applicable to multi-instance data where the distribution for each instance-level class belongs to an expectation-parametrized exponential family. We showed that the models can easily be applied to non-vector data such as sequences. This extends the set of problems to which multi-instance algorithms can be applied to many types of structured instance-level data. For example, it would be straightforward to apply MIMMs to bags of graphs using exponential family random graph models [16] as the instance-level class distributions.

The MIMM framework can potentially be extended to incorporate domain knowledge regarding the generative process. For example, we may have prior knowledge on the distribution of the number of positive instances in a positive bag. Computational intractibility quickly arises in variants such as this where the independence assumption is abandoned, as the E-step involves a sum over all possible assignments to the bag labels, which is exponential in the number of instances in the bags. Approximate approaches such as MCMC sampling or variational inference in such variants are potential avenues for future research.

References

[1] S Andrews, I Tsochantaridis, and T Hofmann. Support vector machines for multiple-instance learning. In *NIPS 15*, pages 561–568, 2003.

[2] V. Castelli and T.M. Cover. On the exponential value of labeled samples. *Pattern Recognition Letters*, 16(1):105–111, 1995.

[3] L. De Raedt. Attribute-value learning versus inductive logic programming: The missing links. In *ICILP 8*, volume 1446 of *LNAI*, pages 1–8, 1998.

[4] A.P. Dempster, N.M. Laird, D.B. Rubin, et al. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B*, 39(1):1–38, 1977.

[5] T. Dietterich, R. Lathrop, and T. Lozano-Perez. Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence*, 89(1-2):31–71, 1997.

[6] J. Foulds and E. Frank. A review of multi-instance learning assumptions. *Knowledge Engineering Review*, 25(1):1–25, 2010.

[7] J. Foulds and E. Frank. Speeding up and boosting diverse density learning. In *Proc 13th International Conference on Discovery Science*, volume 6332 of *LNAI*, pages 102–116. Springer, 2010.

[8] C. Fraley and A.E. Raftery. Bayesian regularization for normal mixture estimation and model-based clustering. *Journal of Classification*, 24(2):155–181, 2007.

[9] H.P. Kriegel, A. Pryakhin, and M. Schubert. An EM-approach for clustering multi-instance objects. In *PAKDD 10*, pages 139–148, 2006.

[10] O. Maron. *Learning from ambiguity*. PhD thesis, Massachusetts Institute of Technology, 1998.

[11] O. Maron and T. Lozano-Pérez. A framework for multiple-instance learning. In *NIPS 10*, pages 570–576, 1998.

[12] A. Ng and M. Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In *NIPS 14*, pages 841–848, 2002.

[13] R. A. Redner and H. F. Walker. Mixture densities, maximum likelihood and the EM algorithm. *SIAM Review*, 26(2):195–239, 1984.

[14] D.B. Rubin. Inference and missing data. *Biometrika*, 63(3):581, 1976.

[15] P. Viola, J. Platt, and C. Zhang. Multiple instance boosting for object detection. In *NIPS 18*, pages 1417–1424, 2006.

[16] S. Wasserman and P. Pattison. Logit models and logistic regressions for social networks: I. An introduction to Markov graphs and p^*. *Psychometrika*, 61:401–425, 1996.

[17] N. Weidmann, E. Frank, and B. Pfahringer. A two-level learning method for generalized multi-instance problems. In *ECML 14*, pages 468–479, 2003.

[18] I. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques, 2nd Ed.* Morgan Kaufmann, San Francisco, 2005.

[19] X. Xu. Statistical learning in multiple instance problems. Master's thesis, University of Waikato, 2003.

[20] X. Xu and E. Frank. Logistic regression and boosting for labeled bags of instances. In *PAKDD 8*, pages 272–281, 2004.

[21] S.H. Yang, H. Zha, and B. Hu. Dirichlet-bernoulli alignment: A generative model for multi-class multi-label multi-instance corpora. In *NIPS 22*, pages 2143–2150, 2009.

[22] Q. Zhang and S. Goldman. EM-DD: An improved multiple-instance learning technique. In *NIPS 14*, pages 1073–1080, 2002.

[23] Y. Zhang, A.C. Surendran, J.C. Platt, and M. Narasimhan. Learning from multi-topic web documents for contextual advertisement. In *SIGKDD 14*, pages 1051–1059, 2008.

[24] Z.H. Zhou and J.M. Xu. On the relation between multi-instance learning and semi-supervised learning. In *ICML 24*, pages 1167–1174, 2007.

A Derivation of the EM Algorithm for MIMMs

In this appendix, we show the derivation of the EM Algorithm for multi-instance mixture models. The reader is referred back to Section 4 for the notation. This derivation is similar to the derivation for the EM algorithm for mixtures of exponential family distributions given in [13].

A.1 E-step By linearity of expectation, $Q(\theta;\theta^k) = E[\sum_{i=1}^{N_B} L_c(\theta;\mathcal{X}_i,y_i,\mathcal{Z}_i)|\mathcal{X},Y,\theta^k]$ can be written as $\sum_{i=1}^{N_B} E[L_c(\theta;\mathcal{X}_i,y_i,\mathcal{Z}_i)|\mathcal{X}_i,y_i,\theta^k]$. So to compute $Q(\theta;\theta^k)$, we can consider each bag separately. We make use of the following lemma (see Appendix B for a proof).

LEMMA A.1. *Suppose V is a discrete random variable, $q \in V$, $P(V=q) \neq 1$, $f(v)$ is a real-valued function on V, and $E[f(v)]$ exists. Then*

$$E[f(v)|V \neq q] = \frac{E[f(v)] - P(V=q)f(q)}{1-P(V=q)} \,.$$

For a positive bag B_i, we can use Lemma A.1 to compute the expected complete-data log-likelihood $E[L_c(\theta;\mathcal{X}_i,y_i,\mathcal{Z}_i)|\mathcal{X}_i,\oplus,\theta^k] = E[L_c(\theta;\mathcal{X}_i,y_i,\mathcal{Z}_i)|\mathcal{X}_i,Z_i \neq 0,\theta^k]$

$$= \frac{E[L_c(\theta;\mathcal{X}_i,\mathcal{Z}_i)|\mathcal{X}_i,\theta^k] - \alpha_i L_c(\theta;\mathcal{X}_i,\mathcal{Z}_i=0)}{1-\alpha_i} \,.$$

Note that the complete-data log-likelihood for a $\mathcal{Z}_i$ assignment inconsistent with the bag label y_i is $\ln(0)$. In calculating the expected log-likelihood, we assume $0 \times \ln(0)$ is 0. The expected complete-data log-likelihood is trivially computed for negative bags, as the bag label implies that all instances in the bag are negative. The sum of the expected log-likelihoods for all negative bags is

$$\sum_{i=1}^{N^-} \ln P_\theta(\mathbf{x}_i^-|z_i=0) + N^- \ln(1-\pi_p) \,.$$

Adding all of the expected log-likelihood terms over all bags gives Equation 4.1.

A.2 M-step We now derive the M-step updates for the MIMM's parameters. First we shall maximize the Q function with respect to π_p. Regardless of the form of the mixture components, in anticipation of taking the derivative with respect to π_p the terms of interest in the expected log-likelihood are:

$$\sum_{i=1}^{N^+} \frac{1}{1-\alpha_i}\left(\sum_{j=1}^{S_i}\Big(\gamma_{ij}\ln\pi_p + (1-\gamma_{ij})\ln(1-\pi_p)\Big) - \alpha_i S_i \ln(1-\pi_p)\right) + N^- \ln(1-\pi_p) \,.$$

Let $\omega_{ij} = \frac{\gamma_{ij}}{1-\alpha_i}$ for instance j of positive bag i. Now taking the derivative with respect to π_p and setting to zero:

$$0 = \sum_{i=1}^{N^+} \frac{1}{1-\alpha_i}\left(\sum_{j=1}^{S_i}\left(\frac{\gamma_{ij}}{\pi_p} - \frac{1-\gamma_{ij}}{1-\pi_p}\right) + \frac{\alpha_i S_i}{(1-\pi_p)}\right) - \frac{N^-}{1-\pi_p}$$

$$0 = \sum_{i=1}^{N^+} \frac{1}{1-\alpha_i}\left(\sum_{j=1}^{S_i}\Big((1-\pi_p)\gamma_{ij} - \pi_p(1-\gamma_{ij})\Big) + \pi_p\alpha_i S_i\right) - \pi_p N^-$$

$$\pi_p\left(\sum_{i=1}^{N^+} \frac{1}{1-\alpha_i}\left(\sum_{j-1}^{S_i}\Big(\gamma_{ij} + 1 - \gamma_{ij}\Big) - \alpha_i S_i\right)\right) + N^-)$$

$$= \sum_{i=1}^{N^+} \frac{1}{1-\alpha_i} \sum_{j=1}^{S_i} \gamma_{ij}$$

$$\pi_p = \frac{\sum_{i=1}^{N^+} \frac{1}{1-\alpha_i} \sum_{j=1}^{S_i} \gamma_{ij}}{\sum_{i=1}^{N^+} \frac{S_i(1-\alpha_i)}{1-\alpha_i} + N^-} = \frac{\sum_{i=1}^{N^+} \frac{1}{1-\alpha_i} \sum_{j=1}^{S_i} \gamma_{ij}}{\sum_{i=1}^{N^+} S + N^-}$$

$$= \frac{\sum_{i=1}^{N^+} \sum_{j=1}^{S_i} \omega_{ij}}{\sum_{i=1}^{N^+} S + N^-}$$

We assume each instance-level class $c \in \{+,-\}$ has an exponential family distribution with natural parameterization $p^c(\mathbf{x}|\theta^c) = a(\theta^c)^{-1}b(\mathbf{x})e^{\theta^{c\intercal}t(\mathbf{x})}$ and expectation parameterization $p^c(\mathbf{x}|\phi^c) = a(\theta^c(\phi))^{-1}b(\mathbf{x})e^{\theta^c(\phi)^\intercal t(\mathbf{x})}$, i.e. $\theta^c \to \phi^c = E[t(\mathbf{x})|\theta]$ is a bijection. In anticipation of taking the derivative with respect to θ^+, the relevant terms of the Q function are

$$\sum_{i=1}^{N^+}(\frac{1}{1-\alpha_i})\sum_{j=1}^{S_i}\Big(- \gamma_{ij}\ln a^+(\theta^+(\phi^+)) + \gamma_{ij}\theta^+(\phi^+)^\intercal t^+(\mathbf{x}_{ij})\Big) .$$

Taking the derivative with respect to θ^+, and setting to zero, we obtain

$$0 = \sum_{i=1}^{N^+}(\frac{1}{1-\alpha_i})\sum_{j=1}^{S_i}\Big(- \gamma_{ij}\phi^+ + \gamma_{ij}t^+(\mathbf{x}_{ij})\Big)$$

where we make use of the well-known property of exponential family distributions that the log-partition function is the cumulant generating function, i.e. the first derivative of the log-partition function $\triangledown \ln a(\theta)$ is $E_{q(\mathbf{x}|\theta)}[t(\mathbf{x})]$, which in our case equals ϕ^+. Rearranging,

$$\phi^+\sum_{i=1}^{N^+}(\frac{1}{1-\alpha_i})\sum_{j=1}^{S_i}\gamma_{ij} = \sum_{i=1}^{N^+}(\frac{1}{1-\alpha_i})\sum_{j=1}^{S_i}\gamma_{ij}t^+(\mathbf{x}_{ij}) .$$

Solving for ϕ^+, the M-step update for ϕ^+ is

$$\phi^+ = \frac{\sum_{i=1}^{N^+}\sum_{j=1}^{S_i}\omega_{ij}t^+(\mathbf{x}_{ij})}{\sum_{i=1}^{N^+}\sum_{j=1}^{S_i}\omega_{ij}} .$$

To find the M-step update for ϕ^-, the relevant terms of the Q function are

$$\begin{aligned}&\sum_{i=1}^{N^+}(\frac{1}{1-\alpha_i})\Bigg(\sum_{j=1}^{S_i}\big(- (1-\gamma_{ij})\ln a^-(\theta^-(\phi^-))\\ &\quad + (1-\gamma_{ij})\theta^-(\phi^-)^\intercal t^-(\mathbf{x}_{ij})\big)\\ &\quad - \alpha_i\sum_{j=1}^{S_i}\big(- \ln a^-(\theta^-(\phi^-)) + \theta^-(\phi^-)^\intercal t^-(\mathbf{x}_{ij})\big)\Bigg)\\ &\quad + \sum_{j=1}^{N^-}\Big(- \ln a^-(\theta^-(\phi^-)) + \theta^-(\phi^-)^\intercal t^-(\mathbf{x}_j^-)\Big)\end{aligned}$$

Taking the derivative with respect to θ^- and setting to zero:

$$\begin{aligned}0 =& \sum_{i=1}^{N^+}(\frac{1}{1-\alpha_i})\Bigg(\sum_{j=1}^{S_i}\big(- (1-\gamma_{ij})\phi^- + (1-\gamma_{ij})t^-(\mathbf{x}_{ij})\big)\\ & - \alpha_i\sum_{j=1}^{S_i}\big(\phi^- + t^-(x_{ij})\big)\Bigg) + \sum_{j=1}^{N^-}\Big(- \phi^- + t^-(\mathbf{x}_j^-)\Big)\end{aligned}$$

By rearranging, we obtain

$$\begin{aligned}&\phi^-\Bigg(\sum_{i=1}^{N^+}(\frac{1}{1-\alpha_i})\sum_{j=1}^{S_i}(1-\gamma_{ij}-\alpha_i) + N^-\Bigg)\\ &\quad = \sum_{i=1}^{N^+}(\frac{1}{1-\alpha_i})\sum_{j=1}^{S_i}(1-\gamma_{ij}-\alpha_i)t^-(\mathbf{x}_{ij}) + \sum_{j=1}^{N^-}t^-(\mathbf{x}_j^-)\end{aligned}$$

and finally solving for ϕ^-, the M-step update for ϕ^- is

$$\begin{aligned}\phi^- =& \frac{\sum_{i=1}^{N^+}(\frac{1}{1-\alpha_i})\sum_{j=1}^{S_i}(1-\gamma_{ij}-\alpha_i)t^-(\mathbf{x}_{ij}) + \sum_{j=1}^{N^-}t^-(\mathbf{x}_j^-)}{\sum_{i=1}^{N^+}(\frac{1}{1-\alpha_i})\sum_{j=1}^{S_i}(1-\gamma_{ij}-\alpha_i) + N^-}\\ =& \frac{\sum_{i=1}^{N^+}\sum_{j=1}^{S_i}(1-\frac{\gamma_{ij}}{1-\alpha_i})t^-(\mathbf{x}_{ij}) + \sum_{j=1}^{N^-}t^-(\mathbf{x}_j^-)}{\sum_{i=1}^{N^+}\sum_{j=1}^{S_i}(1-\frac{\gamma_{ij}}{1-\alpha_i}) + N^-}\\ =& \frac{\sum_{i=1}^{N^+}\sum_{j=1}^{S_i}(1-\omega_{ij})t^-(\mathbf{x}_{ij}) + \sum_{j=1}^{N^-}t^-(\mathbf{x}_j^-)}{\sum_{i=1}^{N^+}\sum_{j=1}^{S_i}(1-\omega_{ij}) + N^-} .\end{aligned}$$

B Proof of Lemma A.1

$$\begin{aligned}E[f(v)|V \neq q] &= \sum_{v\in V} f(v)P(V=v|V\neq q)\\ &= \sum_{v\in V} f(v)\frac{P(V=v, V\neq q)}{P(V\neq q)}\\ &= \frac{1}{1-P(V=q)}\sum_{v\in V} f(v)P(V=v,V\neq q)\\ &= \frac{1}{1-Pr(V=q)}\Bigg(\sum_{v\in V} f(v)P(V=v)\\ &\quad -f(q)P(V=q)\Bigg)\\ &= \frac{E_{P(V)}[f(v)] - f(q)P(V=q)}{1-P(V=q)}\end{aligned}$$