Context-Aware MIML Instance Annotation: Exploiting Label Correlations With Classifier Chains

Citation	Briggs, F., Fern, X. Z., & Raich, R. (2015). Context-aware MIML instance annotation: exploiting label correlations with classifier chains. Knowledge and Information Systems, 43(1), 53-79. doi:10.1007/s10115-014-0781-8
DOI	10.1007/s10115-014-0781-8
Publisher	Springer
Version	Accepted Manuscript
Terms of Use	http://cdss.library.oregonstate.edu/sa-termsofuse

Under consideration for publication in Knowledge and Information Systems

Context-Aware MIML Instance Annotation: Exploiting Label Correlations With Classifier Chains

Forrest Briggs, Xiaoli Z. Fern, and Raviv Raich

School of Electrical Engineering & Computer Science, Oregon State University, Corvallis OR, USA, 97331-5501

Abstract. In multi-instance multi-label (MIML) instance annotation, the goal is to learn an instance classifier while training on a MIML dataset, which consists of bags of instances paired with label sets; instance labels are not provided in the training data. The MIML formulation can be applied in many domains. For example, in an image domain, bags are images, instances are feature vectors representing segments in the images, and the label sets are lists of objects or categories present in each image. Although many MIML algorithms have been developed for predicting the label set of a new bag, only a few have been specifically designed to predict instance labels. We propose MIML-ECC (ensemble of classifier chains), which exploits bag-level context through label correlations to improve instance-level prediction accuracy. The proposed method is scalable in all dimensions of a problem (bags, instances, classes, and feature dimension), and has no parameters that require tuning (which is a problem for prior methods). In experiments on two image datasets, a bioacoustics dataset, and two artificial datasets, MIML-ECC achieves higher or comparable accuracy in comparison to several recent methods and baselines.

Keywords: multiple instance, multi-label, MIML, instance annotation, classifier chain

1. Introduction

The most common formulation of supervised classification is single-instance single-label (SISL), where the training data consists of feature vectors (instances) paired with single labels. The goal is to predict the label for a new instance.

Received Dec 22, 2013
Revised May 03, 2014
Accepted Jul 11, 2014

SVMs, logistic regression, and decision trees are for SISL. Multi-instance multi-label (MIML) learning is a framework for supervised classification, where the dataset is represented as a collection of bags of instances, paired with sets of labels. For example, in an image domain, a bag is an image, the instances in the bag are feature vectors describing regions, and the label set for a bag indicates which objects or categories the image contains. MIML has been applied to to image, text [39, 36, 18, 26], audio [5], and video [31] domains. There are many algorithms that train a classifier on a MIML dataset to predict the label set for a new bag (e.g., the original formulation of MIML [41]).

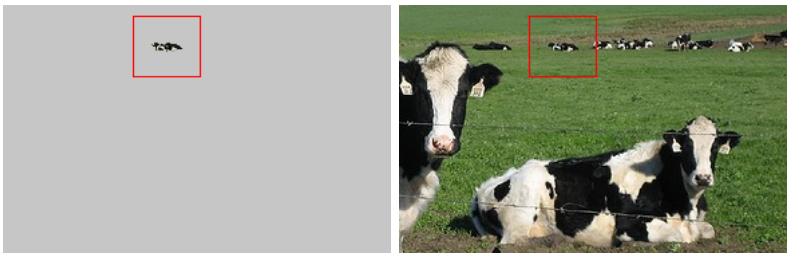
MIML instance annotation is a recent and little-studied problem for supervised classification. In contrast with most prior work on MIML, instance annotation aims to train a classifier on a MIML dataset to **predict the instance labels**. For example, we train a classifier on images paired with sets of objects they contain, then predict the class label for each region of a new image. The main advantage of MIML instance annotation compared to SISL is that it typically requires less human effort to provide bag label sets than to label instances. For example, images tagged with label sets are abundant, whereas SISL data is limited (there are not many images labeled at the pixel-level).

MIML instance annotation differs from the traditional MIML problem of label set prediction (e.g., M^3MIML [37]), and multi-label classification (MLC, e.g., binary relevance). In particular, it is commonly assumed that each instance only belongs to one class, thus the predictions to be made are single labels for instances, not label sets. An appropriate objective for MIML instance annotation is to maximize instance-level accuracy (the fraction of correctly classified instances). However, it is not possible to train a model that directly optimizes accuracy on the training data, because instance labels are not available for training.

Instance annotation problems for images have been widely explored. For example, Yang et al. [34] used structural SVMs with latent variables representing object labels and locations, and learned from image label sets. Sometimes it is possible to modify a MIML or MLC algorithm that is designed for label set prediction, to predict instance labels. The problem with this approach is that the model is optimized for label set accuracy, not instance accuracy. However, to our knowledge only two prior studies have specifically considered the general domain-independent MIML instance annotation problem [3, 4]. Briggs et al. [3, 4] proposed rank-loss Support Instance Machines (SIM), a collection of SVM-style algorithms that learn a linear instance classifier by minimizing a rank loss objective on bag-level labels.

Prior work [3, 4] has observed that the rank-loss SIM algorithms, as well as several other baseline methods, achieve lower accuracy for inductive classification of instances (predicting instance labels for previously unseen bags) in comparison to transductive classifications (predicting instance labels for bags with known label sets). We hypothesize that one way to improve the performance of inductive classification is to exploit the contextual information provided by other instances in the same bag.

Figure 1a illustrates the difficulty of instance annotation without context. The region of pixels inside the red box is an instance. A MIML instance annotation classifier might be asked to predict the class label of this instance. Without the context provided by the rest of the image, it is hard to classify, even for a human. Figure 1b shows the rest of the image. With this context available, it is easier to recognize the instance. Figure 1a illustrates how inductive MIML



(a) Without context (b) With context

Fig. 1. Inductive instance annotation with and without context – "What class is the region of pixels inside the red box?" This image is from the VOC12 dataset.

instance annotation is posed in prior work [3, 4]. It is not as important to use the context provided by other instances in the same bag for transductive classification, because the bag label set is already known, and provides a similar kind of context. Consider the same example in Fig. 1a. If we know that the image contains labels "cow" and "grass," we do not need to see the rest of the image to conclude that the label for this instance should be "cow."

This paper proposes a new algorithm for MIML instance annotation designed to improve inductive instance accuracy by exploiting the context provided by other instances in the same bag. In particular, we capture the context by modeling label correlations in the bag label set. The proposed method (Sec. 4) is a multi-instance multi-label ensemble of classifier chains, called MIML-ECC. The classification algorithm selects the maximum a posteriori (MAP) instance label as estimated by the ensemble, and the training algorithm is closely related to EM and the Constrained Concave-Convex Procedure (CCCP) [35]. Training is asymptotically efficient in all dimensions of a problem (number of bags, instances, classes, and feature dimension). Experiments (Sec. 5) on two image datasets, a bioacoustics dataset, and artificial datasets show that MIML-ECC achieves higher accuracy than several recent methods and baselines, including Hamming, rank, and ambiguous-loss SVMs, and comparable accuracy to a recent graphical model. Further experiments show that the chain structure outperforms binary relevance, and an ensemble of chains outperforms a single chain. To gain a better understanding of how MIML-ECC exploits correlation, we present experiments with artificial datasets where the degree of correlation between classes is controlled.

2. Problem Statement

For training, we are given a MIML dataset consisting of n bags paired with their corresponding label sets $\{(B_1, Y_1), \ldots, (B_n, Y_n)\}$, where B_i is a bag, $Y_i \subseteq \mathcal{Y} = \{1, \ldots, c\}$ is its label set, and c is the total number of classes. Each bag B_i contains n_i instances, i.e., $B_i = \{\mathbf{x}_{i1}, \ldots, \mathbf{x}_{in_i}\}, \mathbf{x} \in \mathcal{X} = \mathbb{R}^d$.

We assume that each instance $\mathbf{x}$ in B_i has a single label $y \in \mathcal{Y}$. The instance labels are not available in the training data; and we only have ambiguous information about them provided through the bag label sets. Our goal is to learn a

Table 1. Frameworks for supervised classification

Framework	Training Dataset	Classifier
SISL	$(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$	$y = f(\mathbf{x}) : \mathcal{X} \to \mathcal{Y}$
MIL	$(B_1, y_i), \ldots, (B_n, y_n)$	$y = F(B) : 2^{\mathcal{X}} \to \{0, 1\}$
MLC	$(\mathbf{x}_1, Y_1), \ldots, (\mathbf{x}_n, Y_n)$	$Y = f(\mathbf{x}) : \mathcal{X} \to 2^{\mathcal{Y}}$
MIML	$(B_1, Y_1), \ldots, (B_n, Y_n)$	$Y = F(B) : 2^{\mathcal{X}} \to 2^{\mathcal{Y}}$
ALC/SLL	$(\mathbf{x}_1, Y_1), \ldots, (\mathbf{x}_n, Y_n)$	$y = f(\mathbf{x}) : \mathcal{X} \to \mathcal{Y}$

classifier that predicts instance labels while training only on the bag-level label sets.

Instance annotation can be applied in both transductive and inductive modes, which differ in what information is available at the classification stage. The transductive classifier is defined as:

$$y = f(\mathbf{x}, B, Y) : \mathcal{X} \times 2^{\mathcal{X}} \times 2^{\mathcal{Y}} \to \mathcal{Y} \tag{1}$$

The notation $2^{\mathcal{X}}$ indicates the space of possible bags, $2^{\mathcal{Y}}$ indicates the space of label sets, and $f(\mathbf{x}, B, Y)$ indicates that we are given all of the instances in a bag B, its label set Y, and the goal is to predict the label y for a specific instance $\mathbf{x}$ in B. For example, in the transductive mode, the prediction task could be: given an image and the list of classes it contains, predict the class of a particular segment in the image.

The inductive mode classifies an instance without the bag label set given. For example, in the inductive mode, the prediction task could be: given an image, predict the class of a particular segment in the image. Prior work [3] on MIML instance annotation formulates the inductive classifier as $f(\mathbf{x}) : \mathcal{X} \to \mathcal{Y}$, which ignores any context from the bag containing $\mathbf{x}$. We instead formulate the inductive classifier as

$$y = f(\mathbf{x}, B) : \mathcal{X} \times 2^{\mathcal{X}} \to \mathcal{Y} \tag{2}$$

The difference is that when classifying an instance $\mathbf{x}$, we know that it is part of a bag B, and can use the contextual information of B to improve the prediction.

Related Problems There are many other formulations of supervised classification that are related to MIML instance annotation. The main difference between these frameworks is the structure of training data (instance or bag, single- or multi-label), and the type of prediction it makes (instance-level or bag-level, single or multi-label). Refer to Table 1 for a statement of the training data and inductive classifier in each framework.

The most common supervised classification formulation is single-instance single-label (SISL). Most standard methods such as support vector machines, decision trees, and logistic regression are for SISL. Multiple-instance learning (MIL) is a framework where the training data consists of bags of instances paired with a single binary label, and the classifier maps bags to binary labels. Multi-label classification (MLC) [25] pairs single instances with sets of labels, and the goal is to predict a label set given a new instance.

Ambiguous label classification (ALC) [8] and superset label learning (SLL) [19] have the same structure of training data as MLC, but assume only one label in the set is correct and the rest are "distractors." The goal is to learn a classifier to predict a single label for a new instance. MIML instance annotation can be reduced to ALC/SLL by pairing each instance with its bag label set. However, this reduction can be undesirable as it discards the context of the bag.

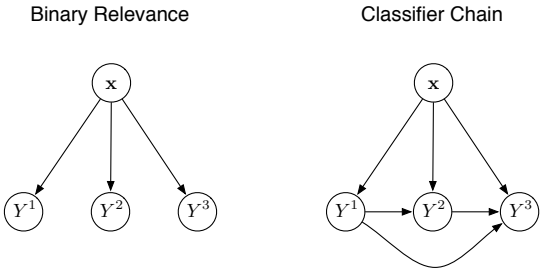


Fig. 2. Graphical models for binary relevance and classifier chians.

3. Background

A key observation motivating our approach is that the context provided by a bag's label set is useful for classifying instances. In the previous example, knowing that there is "grass" in the image can help for predicting the label "cow" for the given instance, because the labels "cow" and "grass" are correlated. A natural way to exploit such context is to follow a classifier-chain approach, which has been previously developed for MLC to exploit label correlation. Below we begin with a review of classifier chains for MLC. We then discuss some design patterns in MIL and MIML algorithms that learn an instance-level model from bag-level labels, which provide inspiration for our algorithm.

3.1. Classifier Chains for Multi-Label Classification

Recall the setup for MLC: Given an instance $\mathbf{x}$, we denote its label set Y as a binary vector: $Y = [Y^1, \ldots, Y^c]$, where $Y^j = 1$ if the label set for instance $\mathbf{x}$ contains class j.

Binary relevance is an algorithm for MLC, which builds one binary model for each class, and treats the classes independently. Binary relevance uses a binary SISL classifier to model $P(Y^j|\mathbf{x})$, for $j = 1, \ldots, c$. Figure 2 the graphical model for binary relevance.

Originally introduced for MLC, classifier chains [25] exploit label correlation by building a chain of binary classifiers. We use $Y^{1:j-1} = [Y^1, \ldots, Y^{j-1}]$ to refer to the first $j-1$ elements of Y. The key idea of classifier chains is to use a chain factorization of the conditional joint distribution of Y:

$$P(Y|\mathbf{x}) = P(Y^1|\mathbf{x}) \prod_{j=2}^{c} P(Y^j|\mathbf{x}, Y^{1:j-1}) \tag{3}$$

During training, one binary model $P(Y^j|\mathbf{x}, Y^{1:j-1})$ is learned for each class j, which depends on $\mathbf{x}$, and all of the preceding classes $1, \ldots, j-1$. Let $\oplus$ denote vector concatenation. The basic training algorithm is:

MLC Probabilistic Classifier Chain – Train

---

for $j = 1, \ldots, c$:
  $\mathcal{D}_j = \{\ldots, (\mathbf{x}_i \oplus Y_i^{1:j-1}, Y_i^j), \ldots\}_{i=1}^n$
  train classifier $P(Y^j|\mathbf{x}, Y^{1:j-1})$ on $\mathcal{D}_j$

For each class j, a binary supervised classification problem $\mathcal{D}_j$ is created (this is a standard SISL problem, not an MLC problem). This 2-class problem has n instances like the original MLC problem. Each instance consists of the original feature vector $\mathbf{x}_i$ concatenated with part of the corresponding label vector $[Y_i^1, \ldots, Y_i^{j-1}]$, and paired with the binary label Y_i^j. The binary model for class j, namely $P(Y^j|\mathbf{x}, Y^{1:j-1})$, can be learned using any binary probabilistic classifier, e.g., logistic regression or Random Forest (RF) [2].

To classify a new instance $\mathbf{x}$ with a probabilistic classifier chain, one can evaluate $P(Y|\mathbf{x})$ for all 2^c possible label vectors Y, and pick one that minimizes a set-level loss function. However, this approach may be intractable unless c is small. An alternative is to greedily construct a single value of Y. A basic greedy algorithm [9] is:

MLC Probabilistic Classifier Chain – Classify

$Y = []$
for $j = 1, \ldots, c$:
  $Y = Y \oplus I[P(Y^j|\mathbf{x} \oplus Y) > 0.5]$
return Y

In ensembles of classifier chains (ECC) [25], there are multiple chains, each of which is learned as above, but factorizing the classes in a different random order. When classifying with ECC, each chain votes. ECC reduces the sensitivity to the specific order of the chain and is generally observed to improve accuracy over a single chain.

3.2. From Instance to Bag Labels

A central problem in MIL and MIML is that labels are only provided at the bag level. Learning an instance classifier from bag label sets requires an assumption about the relationship between the observed label sets and the hidden instance labels. A common assumption in MIL is that if any instance is positive, the bag label is positive, otherwise it is negative. The corresponding assumption in MIML is that the bag label set is equal to the union of instance labels. Prior algorithms approximate these assumptions using different formulations, e.g., the max model.

Let $f(\mathbf{x})$ be an instance-level score function, and $F(B)$ be a bag-level score function. In the MIL setting, the max model is: $F(B) = \max_{\mathbf{x} \in B} f(\mathbf{x})$, i.e. the bag-level score $F(B)$ is the max over the instance-level scores $f(\mathbf{x})$ on all instances in the bag.

For probabilistic MIL classifiers, the max model has also been called the

Table 2. Summary of notation

Notation	Meaning
$\oplus$	vector concatenation operator
B_i	the ith bag in the training data
Y_i	label set for bag B_i, $Y_i \subseteq \{1, \ldots, c\}$
n	number of bags in the training set
n_i	number of instances in bag B_i
π	a permutation function indicating the order of a chain
$\pi(j)$	the j'th class in some permutation π
$\pi_l(j)$	the j'th class in the permutation for chain l
$Y_i^{\pi_l(j)}$	the j'th bit (0 or 1) of the label set Y_i in order π_l
$Y_i^{\pi_l(1):\pi_l(j-1)}$	the first $j-1$ bits of the label set Y_i in order π_l
$\mathbf{x} \in B_i$	an instance in bag B_i, a vector in $\mathbb{R}^d$
f_{jl}	instance-level score function for class $\pi_l(j)$
F_{jl}	bag-level score function for class $\pi_l(j)$
$\hat{\mathbf{x}}_{ijl}$	support-instance for bag i, chain l, class $\pi_l(j)$
y^k	indicator variable for instance $\mathbf{x}$ in class k

"most-likely-cause estimator" [20],

$$P(y = 1|B, \theta) = \max_{\mathbf{x}_i \in B} p(y_i = 1|\mathbf{x}_i, \theta) \tag{4}$$

Here y is the binary label for bag B, and y_i is the binary label for instance $\mathbf{x}_i$. The equivalent formulation for MIML [37, 3] applies the same principle for each class $j = 1, \ldots, c$:

$$F_j(B) = \max_{\mathbf{x} \in B} f_j(\mathbf{x}) \tag{5}$$

Given a model for connecting bag labels with instance labels, the output of a bag-level classifier can sometimes be expressed as a function of a single instance in the bag or representing the bag. For example, assuming the max model for MIL we have:

$$F(B_i) = \max_{\mathbf{x} \in B_i} f(\mathbf{x}) = f(\hat{\mathbf{x}}_i) \tag{6}$$

$$\hat{\mathbf{x}}_i = \arg\max_{\mathbf{x} \in B_i} f(\mathbf{x}) \tag{7}$$

where $\hat{\mathbf{x}}_i$ is referred to as the support instance (or "witness instance" [1]) for bag B_i. We can define support instances similarly for MIML, except that one support instance is defined for each class and each bag. Alternatives to the max model are discussed in Sec. 6.

Many existing algorithms for MIL (e.g., MI-SVM [1] and EM-DD [38]) and MIML (e.g., SIM [4]) alternate between computing support instances based on a current classifier, and training a SISL classifier on the support instances. Our proposed algorithm follows the same pattern.

4. Proposed Methods

Our goal is to learn a classifier that predicts the label of a given instance, using its feature vector $\mathbf{x}$ and the context provided by the bag B containing $\mathbf{x}$. We propose the MIML-ECC algorithm, which is motivated by the observation that the prediction of whether an instance belongs to a particular class can be influenced by the presence/absence of some other classes in the bag. To capture the label correlation, we assume an ordered chain structure such that whether

an instance belongs to a particular class depends on whether the bag contains classes earlier in the chain. First, we present a training algorithm to learn such a model from MIML data. Then we discuss instance classification in transductive and inductive modes. Table 2 summarizes notation for the proposed method.

4.1. Training

A classifier chain for MLC is a chain of SISL classifiers. At a high level, our method can be viewed as building an ensemble of L chains of MIL classifiers. Each chain $l = 1, \ldots, L$ in the ensemble views the classes $1, \ldots, c$ in a different order π_l, such that $\pi_l(j)$ is the j'th class in the order for chain l. We will use F_{jl} to denote the MIL classifier for the j-th class in chain l, which predicts the presence/absence of class $\pi_l(j)$ in the label set of a bag given the bag and $Y^{\pi_l(1):\pi_l(j-1)}$, the presence/absence information of the first $j-1$ classes in chain l. The training algorithm viewed in terms of MIL classifiers is:

MIML-ECC – Train (Bag-Level View)

Input: MIML dataset $\{(B_1, Y_1), \ldots, (B_n, Y_n)\}$
Output: MIL classifiers F_{jl}

---

for $l = 1, \ldots, L$:
  π_l = random–permutation($[1, \ldots, c]$)
  for $j = 1, \ldots, c$:
    $\mathcal{D}_{jl} = \{\ldots, (B_i \oplus Y_i^{\pi_l(1):\pi_l(j-1)}, Y_i^{\pi_l(j)}), \ldots\}_{i=1}^{n}$
    train MIL Classifier F_{jl} on $\mathcal{D}_{jl}$

Each MIL dataset $\mathcal{D}_{jl}$ constructed in the algorithm pairs the bag B_i (and the context $Y_i^{\pi_l(1):\pi_l(j-1)}$) with one bit of the label vector $Y_i^{\pi_l(j)}$. In a standard MIL formulation, there are only bags of instances, so it is a modification of MIL to allow the context $Y_i^{\pi_l(1):\pi_l(j-1)}$, which is a vector in $\{0,1\}^{j-1}$, to be associated with the bag rather than an instance. However, in practice we simply append this vector to the end of all of the instance features.

Because our goal is ultimately to predict instance labels, we instantiate this template with a MIL classifier that internally builds an instance-level model. The instance-level models are SISL probabilistic classifiers f_{jl} for $j = 1, \ldots, c$ and $l = 1, \ldots, L$. We assume f_{jl} maps the input $\mathbf{x} \oplus Y^{\pi_l(1):\pi_l(j-1)}$ to an output in $[0, 1]$ (as is the case for a RF). Recall that $\mathcal{Y} = \{1, \ldots, c\}$; we encode the label $y \in \mathcal{Y}$ of instance $\mathbf{x}$ with c binary indicator variables $y^1, \ldots, y^c$ where $y^j = I[y = j]$, and interpret $f_{jl} : \mathbb{R}^{d+j-1} \to [0, 1]$ as the posterior probability $P(y^{\pi_l(j)} | \mathbf{x}, Y^{\pi_l(1):\pi_l(j-1)})$. MIL classifiers F_{jl} are obtained from the instance-level classifiers using the max model, taking into account the context $Y^{\pi_l(1):\pi_l(j-1)}$:

$$F_{jl}(B_i \oplus Y^{\pi_l(1):\pi_l(j-1)}) = \max_{\mathbf{x} \in B_i} f_{jl}(\mathbf{x} \oplus Y^{\pi_l(1):\pi(j-1)})$$

Similar to the MIL algorithm EM-DD, and rank-loss SIM for MIML, we define the bag-level model in terms of a support instance. In MIML-ECC, there is a different support instance for each bag, class, *and chain*. The bag-level model

in terms of support instances is

$$F_{jl}(B_i \oplus Y^{\pi_l(1):\pi_l(j-1)}) = f_{jl}(\hat{\mathbf{x}}_{ijl} \oplus Y_i^{\pi_l(1):\pi_l(j-1)})$$
$$\hat{\mathbf{x}}_{ijl} = \arg\max_{\mathbf{x} \in B_i} f_{jl}(\mathbf{x} \oplus Y_i^{\pi_l(1):\pi_l(j-1)})$$

The support instance $\hat{\mathbf{x}}_{ijl}$ is the instance in bag B_i that is most representative of class $\pi_l(j)$, according to the classifiers in chain l.

The MIML-ECC training algorithm alternates K times between updating support instances according to the max model, then training SISL classifiers on binary datasets that pair support instances with bits of the label set. In the first iteration, there are no instance classifiers f_{jl} to compute support instances from, so we start by setting the support instances to the average of the instances in each bag, as in [3, 4]. The instance-level view of the training algorithm is:

MIML-ECC – Train (Instance-Level View)

Input: MIML dataset $\{(B_1, Y_1), \ldots, (B_n, Y_n)\}$
Output: SISL classifiers f_{jl}

---

for $l = 1, \ldots, L$:
  π_l = random–permutation($[1, \ldots, c]$)
  for $k = 1, \ldots, K$:
    if $k = 1$ then:
      for $i = 1, \ldots, n$: for $j = 1, \ldots, c$:
        $\hat{\mathbf{x}}_{ijl} = \frac{1}{n_i} \sum_{\mathbf{x} \in B_i} \mathbf{x}$
    if $k > 1$ then:
      for $i = 1, \ldots, n$: for $j = 1, \ldots, c$:
        $\hat{\mathbf{x}}_{ijl} = \arg\max_{\mathbf{x} \in B_i} f_{jl}(\mathbf{x} \oplus Y_i^{\pi_l(1):\pi_l(j-1)})$
    for $j = 1, \ldots, c$:
        $\mathcal{D}_{jl} = \{\ldots, (\hat{\mathbf{x}}_{ijl} \oplus Y_i^{\pi_l(1):\pi_l(j-1)}, Y_i^{\pi_l(j)}), \ldots\}_{i=1}^{n}$
      train SISL classifier f_{jl} on $\mathcal{D}_{jl}$

Similarities with EM The proposed training algorithm is a heuristic, and is not proven to converge over multiple support instances updates. However, empirically we observe convergent behavior. Note that our training algorithm is closely related to some prior work using support instances with expectation maximization (EM), which we discuss below.

EM-DD [38] is a widely used EM-style algorithm for MIL (single-labeled bags of instances). The "E-step" consists of computing support instances, and the "M-step" maximizes likelihood in a model involving the support instances. EM-DD also uses the max model to define the support instances. The main difference in how support instances are treated in MIML-ECC is that each bag has a different support instance for each class and chain. Recall that MIML-ECC trains SISL classifiers f_{jl} in each iteration. If the base SISL classifier maximizes log-likelihood (e.g., logistic regression), there is a direct correspondence with the M-step of EM-DD. In our implementation of MIML-ECC, f_{jl} is a RF using the Gini split criteria, which greedily minimizes squared-loss $\mathcal{L}_2(y, p) = (y - p)^2$ on the training data [6]. If the entropy split criteria were used instead, the RF

would greedily maximize likelihood. Gini and entropy are very similar for binary problems.

4.2. Classification

We consider a probabilistic framework for instance classification based on the maximum a posteriori (MAP) approach. Our method can be viewed as approximately optimizing the instance prediction accuracy. In the training phase, instance-level binary classifiers $f_{jl}(\mathbf{x} \oplus Y^{\pi_l(1):\pi_l(j-1)})$ are obtained for every class j and chain l. The output of f_{jl} can be considered as an estimate of the posterior $P(y^{\pi_l(j)}|\mathbf{x}, Y^{\pi_l(1):\pi_l(j-1)})$. These binary classifiers can be used directly in the transductive mode, where Y is available. In the inductive mode, Y is not given, so we generate samples of Y by conditioning on all instances in the bag.

4.2.1. Transductive Mode

In the transductive mode, we condition on the bag and its label set, and predict instance labels according to

$$f(\mathbf{x}, B, Y) = \arg\max_{j \in Y} P(y^j|\mathbf{x}, B, Y) = \arg\max_{j \in Y} P(y^j|\mathbf{x}, Y)$$

This prediction rule assumes that bag label set Y provides all of the contextual information that is relevant to predicting the label for $\mathbf{x}$, i.e. the label is conditionally independent of the other instances in the bag B given Y.

During training we introduced random orders π for the purpose of constructing an ensemble. Now we take a Bayesian approach and assume that π is random variable from a uniform prior $P(\pi)$, so each chain in the ensemble corresponds to one i.i.d. sample $\pi_l \sim P(\pi)$ for $l = 1, \ldots, L$. We estimate the probability for instance $\mathbf{x}$ to have label $y = k$ as $P(y^k|\mathbf{x}, Y) = E_\pi[P(y^k|\mathbf{x}, Y, \pi)]$ using L samples, one for each chain in the ensemble:

$$P(y^k|\mathbf{x}, Y) \approx \frac{1}{L} \sum_{l=1}^{L} \sum_{\{j:\pi_l(j)=k\}} P(y^{\pi_l(j)}|\mathbf{x}, Y^{\pi_l(1):\pi_l(j-1)}, \pi_l)$$

The algorithm for classification in the transductive mode is:

MIML-ECC – Classify (Transductive)

Input: instance $\mathbf{x}$, label set Y
Output: label y

---

for $j = 1, \ldots, c : y^j = 0$
for $l = 1, \ldots, L :$
  for $j = 1, \ldots, c :$
    $y^{\pi_l(j)} = y^{\pi_l(j)} + f_{jl}(\mathbf{x} \oplus Y^{\pi_l(1):\pi_l(j-1)})$
$y = \arg\max_{j \in Y} y^j$

4.2.2. Inductive Mode

In the inductive setting, the bag label set is not given, so the posterior required for classification conditions only on the instances from bag B (and not the bag label set). Therefore, we predict the instance label as the class with the highest posterior probability

$$y = f(\mathbf{x}, B) = \arg\max_{j=1,\ldots,c} P(y^j|\mathbf{x}, B) \tag{8}$$

The probability $P(y^j|\mathbf{x}, B)$ is not directly modeled by the instance-level classifiers f_{jl}; instead we estimate this probability by marginalizing $P(y^j|\mathbf{x}, Y, B)$ over the latent variable Y. This process requires a probabilistic model for Y given B, which we develop below. We begin by stating the assumptions of this model.

Assumption 1: There exist one or more chain orders π, such that for an instance $\mathbf{x}$, the indicator variable $y^{\pi(j)}$ for that instance to belong to class $\pi(j)$ is conditionally independent of any other instances in the same bag B, given the first $j-1$ bits of the bag-level label set $Y^{\pi(1):\pi(j-1)}$:

$$\boxed{P(y^{\pi(j)}|\mathbf{x}, B, Y^{\pi(1):\pi(j-1)}, \pi) = P(y^{\pi(j)}|\mathbf{x}, Y^{\pi(1):\pi(j-1)}, \pi)}$$

In other words, $Y^{\pi(1):\pi(j-1)}$ provides all of the context that is useful to decide if $\mathbf{x}$ belongs to class $\pi(j)$, and it is not necessary to consider the rest of the label set, or the other instances in the bag. We give an example of a machine vision problem where this assumption is reasonable in Section 4.3.

For training, we defined the relation between instance labels and bag label sets according to the max model. The max model is also part of our assumptions for inference, although we will rewrite it in probability notation.

Assumption 2: Bag label sets and instance labels are linked via the max model,

$$\boxed{P(Y^{\pi(j)}|B, Y^{\pi(1):\pi(j-1)}, \pi) = \max_{\mathbf{x}\in B} P(y^{\pi(j)}|\mathbf{x}, Y^{\pi(1):\pi(j-1)}, \pi)}$$

Section 3.2 discusses the motivation for this assumption in depth.

Similar to a classifier chain for MLC, the conditional distribution of the bag label set is factored as a chain in the order π as

$$P(Y|B, \pi) = P(Y^{\pi(1)}|B, \pi) \prod_{j=2}^{c} P(Y^{\pi(j)}|B, Y^{\pi(1):\pi(j-1)}, \pi)$$

Recall that Assumption 2 defines the conditional probability for $Y^{\pi(j)}$ in terms of the instance-level probabilities for $y^{\pi(j)}$, while Assumption 1 defines the instance-level probabilities for $y^{\pi(j)}$ in terms of $Y^{\pi(1):\pi(j-1)}$.

We estimate $P(y^j|\mathbf{x}, B)$ by sampling as follows. For a given π, we apply Assumption 1 to obtain

$$\begin{aligned} P(y^{\pi(j)}|\mathbf{x}, B, \pi) &= E_{Y^{\pi(1):\pi(j-1)}|B,\pi}\left[P(y^{\pi(j)}|\mathbf{x}, Y^{\pi(1):\pi(j-1)}, B, \pi)\right] \\ &= E_{Y^{\pi(1):\pi(j-1)}|B,\pi}\left[P(y^{\pi(j)}|\mathbf{x}, Y^{\pi(1):\pi(j-1)}, \pi)\right] \end{aligned} \tag{9}$$

Because π is a permutation, computing $P(y^{\pi(j)}|\mathbf{x}, B, \pi)$ for $j = 1, \ldots, c$ implies computing $P(y^j|\mathbf{x}, B, \pi)$ for all j.

Finally, we average the posterior estimates over multiple samples from a uniform prior on π:

$$P(y^j|\mathbf{x}, B) = E_\pi\big[P(y^j|\mathbf{x}, B, \pi)\big] \tag{10}$$

As in the transductive mode, each chain in the ensemble gives one sample of $\pi_l \sim P(\pi)$ to estimate the expectation. The inductive classification algorithm is:

MIML-ECC – Classify (Inductive)

Input: bag $B = \{\mathbf{x}_1, \ldots, \mathbf{x}_{n_i}\}$
Output: instance labels $y_1, \ldots, y_{n_i}$

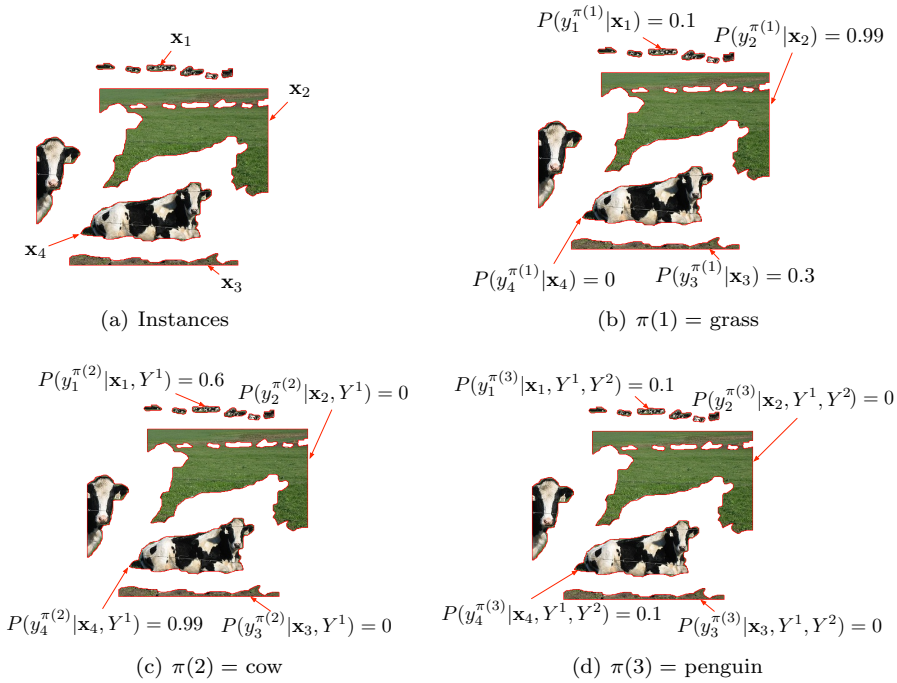
```
01: for i = 1,...,n_i : for j = 1,...,c :
02:       y_i^j = 0
03: for l = 1,...,L :
04:       Y = []
05:       for j = 1,...,c :
06:             for i = 1,...,n_i :
07:                   y_i^{π_l(j)} = y_i^{π_l(j)} + f_jl(x_i ⊕ Y)
08:             p_j = max_{i=1,...,n_i} f_jl(x_i ⊕ Y)
09:             Y = Y ⊕ Bernoulli(p_j)
10: for i = 1,...,n_i :
11:       y_i = arg max_{j=1,...,c} y_i^j
```

Line 7 updates the estimate of $y_i^{\pi_l(j)}$ based on one sample of the expectation (9). Line 8 applies the max model (Assumption 2). In lines 4 through 8, the pseudocode variable Y stores $Y^{\pi_l(1):\pi_l(j-1)}$. In Section 3.1, we discussed the basic greedy algorithm for classifier chains in MLC. In that algorithm, bits of the label set are added deterministically, depending on whether a probability is above or below 0.5. Line 9 serves an analogous purpose in MIML-ECC, but instead the bits are generated randomly based on a probability. Specifically, line 9 samples $Y^{\pi_l(j)}$ from a $Bernoulli(p_j)$ distribution, and appends it to the current label vector.

4.3. Example of Inductive Classification

To clarify how MIML-ECC classifies in the inductive mode, we provide a hypothetical example of object recognition in an image domain. Suppose an image is segmented into instances $\mathbf{x}_1, \ldots, \mathbf{x}_4$ as in Fig. 3a, and there are three classes: grass, cow, and penguin. To simplify the example, we consider a single chain ordered so that $\pi(1) =$ grass, $\pi(2) =$ cow, and $\pi(3) =$ penguin. Classification might proceed as follows:

– The instance-level score function f_1 predicts the indicator $y_i^{\pi(1)}$ for each instance to belong to class $\pi(1) =$ grass, given only the instance feature vector $\mathbf{x}_i$ (Fig. 3b). Instance $\mathbf{x}_1$ is very small, and with no context it is difficult to determine what it is. However, it is not green, so the score function $f_1(\mathbf{x}_1)$ may be low, e.g., 0.1. The instance $\mathbf{x}_2$ is very likely to be grass based on its color, so we will assume the score function returns 0.99. The instance $\mathbf{x}_3$ looks



(a) Instances (b) $\pi(1) =$ grass (c) $\pi(2) =$ cow (d) $\pi(3) =$ penguin

Fig. 3. A hypothetical example illustrating how MIML-ECC classifies instances in the inductive mode with a single chain in the order $\pi(1) =$ grass, $\pi(2) =$ cow, $\pi(3) =$ penguin.

somewhat like grass, but much less so than $\mathbf{x}_2$; suppose the score function for this instance is 0.3. Instance $\mathbf{x}_4$ does not look at all like grass, so we assume a score for class $\pi(1)$ of 0. This step corresponds to line 7 of the pseudocode (for a single chain).

– The first bit of the label set is sampled as $Y^1 \sim Bernoulli(p_1)$ where p_1 is the max of the instance-level scores for class $\pi(1)$, i.e., $p_1 = \max\{0, 0.1, 0.3, 0.99\}$. Most likely, this will result in a sample $Y^1 = 1$ which corresponds to the result that this bag should have grass in its label set, because at least one instance looks like grass. This step corresponds to lines 8 and 9 of the pseudocode.

– The instance-level score function f_2 predicts the indicator $y_i^{\pi(2)}$ for each instance to belong to class $\pi(2) =$ cow, given the instance feature vector $\mathbf{x}_i$, *concatenated with the bit of the label set that was just sampled*, Y^1 (see Fig. 3c). Instance $\mathbf{x}_1$ would be difficult to identify without context, but given that we already sampled a bit indicating the image contains grass, and the fact that cow and grass are correlated, it receives a higher probability than it might otherwise, e.g., 0.6. Instances $\mathbf{x}_2$ and $\mathbf{x}_3$ do not look at all like a cow, so are assigned probability 0. Instance $\mathbf{x}_4$ is clearly recognizable as a cow, and the context of grass being present reinforces the conclusion that it is a cow, hence the score function assigns it a high probability, e.g., 0.99.

– The second bit of the label set is sampled as $Y^2 \sim Bernoulli(\max\{0, 0, 0.6, 0.99\})$, and most likely $Y^2 = 1$. Hence the prediction is that cow is in the label set.

– The instance-level score function f_3 predicts the indicator $y_i^{\pi(3)}$ for each instance to belong to class $\pi(3) =$ penguin, given the instance feature vector $\mathbf{x}_i$, concatenated with Y^1 and Y^2 (Fig. 3d). Without context, the instances $\mathbf{x}_1$ and $\mathbf{x}_4$ might be given a high probability to be penguin, on the basis of their colors. However, given the context of Y^1 and Y^2, they are given lower probabilities because there is a negative correlation between penguins and grass, and penguins and cows.
– The last bit of the label set Y^3 is sampled, although it is irrelevant to the instance labels. The instance labels are predicated according to the highest scoring class. For example, the label predicted for $\mathbf{x}_1$ is cow, because that class received the highest score while progressing through the chain.

The steps above show how a single chain generates the class-scores for each instance. With multiple chains, the scores from each chain are summed before picking the highest-scoring class.

In this example, it is difficult to tell if the instance $\mathbf{x}_1$ belongs to the class cow or penguin without context, but with the context that grass is in the label set, it is easier to identify as cow. This example is just for the sake of illustration. However, we present results on a similar situation where there are two correlated classes, one of which is easily confused with a third class in Sec. 5.5.

4.4. Asymptotic Complexity

MIML-ECC implemented with RF as the base SISL classifier is asymptotically efficient in all important dimensions of the problem size. The size of a MIML dataset is determined by the number of bags n, the total number of instances in all bags m, the number of classes c, and the instance feature dimension d. MIML-ECC has several parameters that affect its runtime: the number of chains L, the number of trees in each RF T, and the number of support-instance updates K. Note that the runtime to train a RF on a SISL dataset of n instances with feature dimension d is $O(T(\log d)(n \log n))$, and to classify it is $O(\log n)$. It follows from the loop-structure of the pseudocode that the training time for MIML-ECC is

$$O\Big(LKT\big(m(\log n)(\log d) + cn \log n \log(d + c)\big)\Big) \tag{11}$$

An efficient implementation of MIML-ECC classifies all instances in a bag at once, rather than treating each instance classification problem separately, in order to share redundant work. Using this optimization, the classification time is $O(LTc \log n)$ per instance. In Section 5.4 we provide empirical runtime results.

5. Experiments

Our experiments compare MIML-ECC to prior and baseline methods on two vision datasets, an audio dataset, and two artificial datasets. Our experimental setup is identical to the setup used in [4] and [19], hence results are directly comparable (e.g., the same features and folds for cross validation are used). Therefore we report new results for MIML-ECC and baseline methods, and compare to previously reported results from the aforementioned prior work.

Table 3. MIML datasets

Dataset	Classes	Dimension	Bags	Instances
MSRCv2	23	48	591	1,758
VOC 2012	20	48	1,053	4,142
Birdsong	13	38	548	4,998
Carroll	26	16	166	717
Frost	26	16	144	565

5.1. Datasets

The datasets used in our experiments are summarized in Table 3. Datasets have been preprocessed through feature rescaling (which does not affect RF), to improve results for SVM style-classifiers, by the same process in [8, 3, 4].

Vision Datasets We consider two vision datasets, Microsoft Research Cambridge v2 (MSRCv2) [29], and PASCAL Visual Object Recogntion Challenge (2012 "Segmentation") [11]. Both datasets contain images of objects with pixel-level labeling of regions. MSRCv2 provides a single class label for each pixel. VOC provides a segmentation of each image into objects and a label for each object. Here bags are images labeled with a list of objects, instances are objects (regions of pixels), described by a 48-D feature vector. Single-label images are removed to make the learning problem more challenging.

Bioacoustics Dataset This dataset was introduced by [5], applying a MIML formulation for label set prediction to a real-world application of classifying bird song collected in field conditions. Each bag is a 10 second audio recording labeled with the set of species it contains. Each instance is an utterance of bird sound obtained by an automatic segmentation algorithm. This dataset has also been used in work on MIML instance annotation and superset label learning [3, 4, 19]. For instance annotation, [3] introduced two variants of this dataset, "filtered" and "unfiltered." Our experiments use the filtered variant, as does [19].

Artificial Datasets We use the same artificial MIML datasets as [3, 4], which are generated to simulate correlations between labels by using letter correlation in English words. The datasets are generated based on the words in two poems, "Jabberwocky" by Lewis Carroll [7], and "The Road Not Taken" by Robert Frost [13], hence they are referred to as Carroll and Frost. Each bag is a word, its letters are instances, and the bag label set is the union of instance labels. The instance features are sampled randomly from the UCI Letter Recognition dataset [12].

5.2. Prior Methods

We compare MIML-ECC with a number of prior methods that can be applied to MIML instance annotation.

M^{3}MIML Originally intended for label-set prediction, M^{3}MIML is a MIML support-vector machine algorithm, which builds one linear instance-level model per class by minimizing a heuristic relaxation of bag-level hinge loss, and connecting instance labels with bag label sets by the max model. Although not

intended for this purpose, the learned instance-level models can be used for instance annotation.

Rank-loss SIM Rank-loss SIM was introduced by [3], and refers to a class of instance annotation algorithms which learn one linear instance-level model per class by minimizing a bag-level rank-loss objective. Different variants of rank-loss SIM consider different models for connecting bag-level output with instance-level outputs, and apply different procedures for optimizing the rank-loss objective. We consider SIM-Heuristic using a softmax model and SIM-CCCP with the max model, with random Fourier kernel features [23] to achieve nonlinear classification by approximating an RBF kernel. These models are chosen for comparison because they achieved the best accuracy in [4].

CLPL Like the other SVM-style algorithms, Convex Learning from Partial Labels (CLPL) [8] learns one linear instance-level model per class, but uses an ALC formulation instead of MIML. CLPL minimizes a loss function which can be seen as an upper bound to the 0/1 loss on the true-unknown label, which is part of the candidate label set.

LSB-CMM Logistic Stick-Breaking Conditional Multinomial Model (LSB-CMM) [19] is a recent hybrid generative / discriminative graphical model for SLL that have been used (by reduction) to solve the instance annotation problem. In particular, the same Birdsong and MSRCv2 datasets were used in [19] to evaluate its instance annotation accuracy. We compare to the results reported in [19] on these two datasets.

5.3. Experimental Setup

Transductive and Inductive In the transductive mode, there is no cross-validation (the whole dataset is used for training and testing). However, because MIML-ECC is a randomized algorithm, we run 10 repetitions and report the average accuracy $\pm$ the standard deviation over repetitions. Most of the other algorithms we compare to are not randomized, so in the transductive mode there is no uncertainty associated with the accuracy result.

In the inductive mode, we use 10-fold cross validation, except for the VOC dataset, for which there is a pre-specified partition into "train" and "val" sets. Results with 10-fold cross-validation are reported as average accuracy over all folds $\pm$ standard deviation. A different random instantiation of MIML-ECC is used in each fold, so we do not run multiple repetitions on top of cross-validation. However, because there is only one fold for the VOC dataset, we report results $\pm$ standard deviation over 10 repetitions for MIML-ECC (and the randomized baseline method SISL Random Forest) on VOC.

M^{3}MIML, CLPL, and rank-loss SIM-Heuristic/CCCP all build one instance-level model per class $f_j(\mathbf{x})$. In the inductive mode, these models are used to predict an instance label by the rule $f(\mathbf{x}) = \arg\max_{j=1,\ldots,c} f_j(\mathbf{x})$. In the transductive mode, the rule is $f(\mathbf{x}, Y) = \arg\max_{j \in Y} f_j(\mathbf{x})$ (hence when the bag label set Y is known, it is used to constrain the instance-label predictions). This constraint provides some context for instance-label prediction, so one might not expect as much benefit to be had from looking at other instances in the transductive mode.

Table 4. Instance annotation accuracy († – results from [4], ‡ – results from [19])

(a) Transductive accuracy ± standard deviation over 10 repetitions for MIML-ECC and SIM-RF

Algorithm	Carroll	Frost	Birdsong	MSRCv2	VOC	Avg Rank
Proposed Methods						
MIML-ECC ($L = 20, K = 20, T = 100$)	.803 ± .006	.831 ± .004	.779 ± .003	.805 ± .007	.624 ± .004	1.8
Prior Methods						
† CLPL	.672	.688	.742	.678	.598	4.0
† M^{3}MIML	.454	.532	.651	.547	.533	5.0
† SIM-CCCP `max` + kernel	.807	.780	.829	.798	.623	2.2
† SIM-Heuristic `softmax` + kernel	.794	.819	.833	.766	.634	2.0
Baseline Methods						
SIM-RF ($K = 20, T = 100$)	.756 ± .0148	.807 ± .0137	.782 ± .009	.777 ± .010	.619 ± .005	

(b) Inductive accuracy ± standard deviation over 10-fold cross validation or 10 repetitions for VOC

Algorithm	Carroll	Frost	Birdsong	MSRCv2	VOC	
Proposed Methods						
MIML-ECC ($L = 20, K = 20, T = 100$)	.618 ± .059	.646 ± .048	.666 ± .052	.611 ± .038	.430 ± .004	1
Prior Methods						
† CLPL	.464 ± .058	.506 ± .063	.620 ± .038	.431 ± .036	.345	3.6
† M^{3}MIML	.288 ± .041	.313 ± .041	.433 ± .073	.317 ± .055	.396	4.2
† SIM-CCCP `max` + kernel	.618 ± .042	.576 ± .065	.630 ± .040	.519 ± .044	.343	2.6
† SIM-Heuristic `softmax` + kernel	.596 ± .041	.587 ± .066	.642 ± .039	.506 ± .038	.337	2.8
‡ LSB-CMM	–	–	.715	.459	–	
Baseline Methods						
SIM-RF ($K = 20, T = 100$)	.542 ± .059	.562 ± .069	.636 ± .050	.584 ± .042	.437 ± .004	
MIML-ECC ($L = 1, K = 20, T = 2000$)	.530 ± .047	.598 ± .040	.644 ± .044	.580 ± .047	.425 ± .003	
SISL Methods (uses instance labels)						
† SISL SVM (multi-class,linear)	.772 ± .049	.753 ± .038	.772 ± .032	.638 ± .045	.440	
SISL Random Forest ($T = 1000$)	.809 ± .049	.807 ± .076	.805 ± .033	.729 ± .050	.511 ± .002	

Parameter Selection All of the rank-loss SIM algorithms, CLPL, M^{3}MIML, and SISL SVM have a regularization parameter (either λ or C). When random kernel features are used to approximate the RBF kernel, there is also a kernel parameter γ, and a parameter D which controls the approximation accuracy. In prior work, these parameters are optimized post-hoc by a grid search as described in [4]. This means the experiment is run once for each parameter setting in a grid, and the best test accuracy over all parameters is reported. Post-hoc selection is not feasible without using instance labels to compute which parameter setting has the best accuracy, but it has been accepted in prior work on MIML instance annotation because it is an unsolved problem. Results using post-hoc selection can be interpreted as the highest accuracy that can be achieved using an oracle to select meta-parameters. Results listed in Table 4 that are marked with a † are obtained with post-hoc parameter selection.

An important practical advantage of MIML-ECC compared to the above prior methods is that it does not have regularization parameters that must be tuned. Note that MIML-ECC has parameters L,K, and T. The accuracy of the algorithm tends to increase as these parameters increases up to a limit. So the parameter choices primarily depend on the time budget for training and testing. Our experiments set $L = 20, K = 20, T = 100$, which provides a good tradeoff between runtime and accuracy.

LSB-CMM [19] has some parameters which can affect accuracy, but in their experiments these parameters are set to standard values for all datasets.

5.4. Results

Comparison With Prior Methods MIML instance annotation algorithms are evaluated based on accuracy, which is the fraction of correctly classified instances. These experiments compare multiple classifiers on multiple datasets,

so following the recommendations of [10], we summarize results using wins, ties, and losses, and average ranks. Table 4 lists the accuracy and average rank results in transductive and inductive modes. Average ranks are computed by sorting the accuracy of MIML-ECC, and the prior methods M^{3}MIML, CLPL, SIM-Heuristic, and SIM-CCCP on each dataset, then averaging the position in the sorted list over all datasets. We do not include LSB-CMM in the ranking because there are only 2 datasets with comparable results.

In the inductive mode, MIML-ECC ties with SIM-CCCP max with RBF kernel on the Carroll dataset, and wins in all other comparisons. Results are not as decisive in the transductive mode, but MIML-ECC still achieves the best average rank over all datasets. This is consistent with our expectation because the known label sets provide a surrogate for context to the other algorithms.

It should be noted that due to the use of post-hoc selection in experiments for CLPL, M^{3}MIML, SIM-Herustic and SIM-CCCP, they are actually given an unfair advantage compared to MIML-ECC, which does not use the test data ground truth in training or parameter selection.

The comparison with LSB-CMM on two datasets is less conclusive. MIML-ECC outperforms LSB-CMM by a margin of 15.2% on the MSRCv2 dataset, but LSB-CMM is slightly better (by a margin of 4%) on the Birdsong dataset.

Ensemble of Chains vs. Binary Relevance (SIM-RF) MIML-ECC is motivated by the idea that bag-level label correlations captured through the chain structure are useful for predicting instance labels. However, it is possible that the improved performance we observe compared to prior linear/kernel algorithms is not due to exploiting label correlations, but instead to using a RF as the base-classifier. To address this hypothesis, we consider an additional comparison against a baseline that we call SIM-RF, which is the same as MIML-ECC in all details except it does not use a chain or model correlations. SIM-RF is equivalent to running MIML-ECC with one chain ($L = 1$) but omitting all of the concatenation of label set bits, i.e. $\oplus Y^{\pi 1:\pi(j-1)}$. SIM-RF is also equivalent to binary relevance with each class modeled by a MIL classifier which alternates between computing support instances and training an RF on them.

MIML-ECC achieves better accuracy than SIM-RF most of the time. The win-loss count is 4-1 in favor of MIML-ECC for both transductive and inductive modes. The comparison to SIM-RF suggests that the chain structure is actually critical, and the improved performance of MIML-ECC compared to prior methods cannot be attributed only to switching from a linear or kernel SVM classifier to RF.

Single Chain vs. Ensemble of Chains We want to know how much benefit the ensemble provides compared to a single chain. The results we reported so far are obtained with $L = 20, K = 20, T = 100$, i.e., 20 chains and 100 trees and 20 iterations of support instance updates. To understand the impact of using mulitple chains with a fair comparison, we run MIML-ECC with one chain order ($L = 1$), and $K = 20, T = 2000$, so the total number of decision trees that vote on an instance label is the same. Table 4b lists results for 1-chain MIML-ECC in the inductive mode (see Baseline Methods). In this comparison, MIML-ECC with multiple chains achieves higher accuracy on all datasets than MIML-ECC with a single chain. These results suggest that given a fixed time budget, it is better to have multiple chains, each with less trees, than a single chain with more trees. Recall that when predicting instance scores for class j, each chain can only

Table 5. Runtime for training and classification with MIML-ECC, per fold of cross-validation or per repetition (seconds)

Mode	Carroll	Frost	Birdsong	MSRCv2	VOC
Transductive	104.9	84.4	251.8	304.5	798.0
Inductive	69.4	57.8	135.5	202.8	2895.8

use the presence/absence of other classes which come before j in the chain. Using multiple chains with random orders increases the chance that relevant classes are available for use as context (at least in some of the chains).

Comparison to SISL We also consider SISL algorithms, which have an unfair advantage of learning directly from instance labels. Results with these SISL algorithms are presented for the inductive mode as an empirical upper bound on the accuracy that can be achieved on these datasets. For this comparison, we use a SISL RF (with 1000 trees), and refer to prior results from [4] with a SISL multi-class linear SVM.

SISL methods achieve better accuracy in inductive experiments than MIML instance annotation, ALC and SLL (Table 4b), which is expected because they are trained on unambiguously labeled instances. This improved accuracy must be weighed against the greater human effort required to obtain instance labels compared to bag label sets.

Empirical Runtime Table 5 lists empirical runtimes for training plus classification with MIML-ECC (with $L = 20, K = 20, T = 100$), on each dataset, averaged over the number of repetitions or folds of cross-validation. The runtime is on the order of seconds or minutes for all datasets. In our experiments, training is parallelized using threads[1], and classification is done sequentially[2].

5.5. Experiments With Controlled Correlation

MIML-ECC is motivated by improving instance annotation accuracy by exploiting correlation in the label set. In order to gain a better understanding of how correlation affects both MIML-ECC and SIM-RF, we conduct additional experiments in which the correlation between classes is controlled.

Similar to the Carroll and Frost datasets, we obtain instance feature vectors from the UCI Letter Recognition dataset [12]. The UCI Letter dataset is a SISL dataset with 26 classes (one for each letter of the alphabet), and 16-d feature vectors. For the purpose of these correlation experiments, however, we only use subsets of 3 or 4 classes from the original dataset, which are chosen deliberately to illustrate a situation where correlation is expected to be beneficial. In particular, we will consider a setup where there are two correlated classes that are easy to distinguish, and a third class which is hard to distinguish from one of the correlated classes. In order to identify classes that meet these criteria from the

[1] We found it effective to use a pool of threads, with each handling one of the L chains. Within each of these threads, construction of the RF classifiers was parallelized over trees. Support instance updates cannot be parallelized, because they occur sequentially in time.

[2] Code is C++ compiled with GCC 4.0 (most speed optimizations enabled). Experiments ran on a Mac Pro with 2x 2.4 GHz Quad-Core Intel Xeon processor and 16 GB 1066 MHz DDR3 memory, with OS X 10.8.1.

26 available in the UCI Letter Recognition dataset, we consider the confusion matrix for a SISL experiment with the full 26 classes. Training and test sets are formed by splitting the instances from each class randomly into 50% training and 50% test; this split results in 10007 training examples and 9993 test examples. A Random Forest with 100 trees is trained on one set, then used to predict the labels in the other.

From the confusion matrix, we find that H is the lowest-accuracy class, with only 0.389646 probability for H instances to be classified correctly. Instances belonging to class H are most often confused with instances of class X, with probability 0.125341. However, H is easily distinguished from instances of class A; zero instances of H are misclassified as A, and instances of A are misclassified as H with probability 0.0027248. Therefore we will setup a MIML instance annotation experiment with three classes: A, H and X. The correlated classes will be A and H, and X will be uncorrelated. We hypothesize that MIML-ECC will achieve better accuracy when A and H are positively or negatively correlated in the label set, because the context provided by the presence or absence of A can help to differentiate between H and X.

The next step in setting up this experiment is to generate bag label sets in such a way that the occurrence of A and H has correlation related to a parameter ρ. Let the label set for a bag B be $Y = [y^A, y^H, y^X]$, where $y^A, y^H, y^X \in \{0, 1\}$. Will will assume a prior for y^A and y^X of $P(y^A) = \frac{1}{2}, P(y^X) = \frac{1}{2}$. Hence it suffices to generate $y^A \sim Bernoulli(\frac{1}{2})$ and $y^X \sim Bernoulli(\frac{1}{2})$. Finally, to obtain y^H such that y^A and y^H have correlation ρ, we generate

$$y^H \sim Bernoulli(\frac{1}{2}(\rho + 1)y^A + \frac{1}{2}(1 - \rho)(1 - y^A)) \tag{12}$$

A proof of that this process generates y^A and y^H with correlation ρ is given in Appendix 1.

We are now ready to supply the remaining details of the experiment. For each value of $\rho \in \{-1.0, -0.9, \ldots, 0.9, 1.0\}$, we generate 100 bags for training and 100 bags for test with label sets as described above. For each class that is present in a bag's label set, we sample 5 instances from the corresponding class randomly. At each value of ρ, we train MIML-ECC or SIM-RF on the 100-bag training set, then classify all instances in the 100-bag test set. The parameters for MIML-ECC are $L = 10, K = 10, T = 10$, and the parameters for SIM-RF are $T = 100, K = 100$, hence both algorithms generate the same total number of trees. Because this is a random experiment, we use 1000 repetitions at each value of ρ to compute statistics about the results. These experiments are conducted in the inductive mode only.

One issue that can occur with the setup described above is that a bag may be generated with an empty label set, and consequently it will have no instances. Such bags are not valid input for MIML-ECC or SIM-RF. We handle this issue with two different variants of the experiment. One approach is to discard any bag that is generated with an empty label set. This approach has the consequence that the sample correlation coefficient $\hat{\rho}$ between y^A and y^H is not equal to the parameter ρ. Figure 4 shows $\hat{\rho}$ as function of ρ, with empty label sets discarded, estimated by sampling 1,000,000 label sets at each value of ρ. Rejecting empty label sets causes $\hat{\rho}$ to be systematically lower than ρ, although there is still a value of ρ such that $\hat{\rho} \approx 0$. Furthermore, discarding empty label sets means that y^X is not conditionally independent of (y^A, y^H). For example, knowing $y^A = 0$

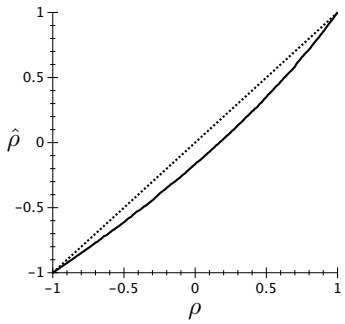


Fig. 4. Sample correlation $\hat{\rho}$ between y^A and y^H as a function of the parameter ρ (empty label sets are discarded). The dotted line is $\hat{\rho} = \rho$.

and $y^H = 0$ implies $y^X = 1$. This kind of relationship is not captured by pairwise correlation.

The second approach we use to address this issue is to inject "noise" instances into each bag, which come from an unknown class that is not accounted for by the label set. Such noise instances realistically occur in machine vision and audio datasets, because automatic segmentation often produces instances that do not belong to any class in the label set. For each randomly generated bag, we add 5 noise instances from the UCI Letter Recognition class W (there is 0 confusion between W and A, H, and X in in both directions). This approach guarantees that all bags have some instances, so it is not necessary to discard bags with empty label sets. In this case, the sample correlation between A and H is not biased, i.e. $\hat{\rho} \approx \rho$ and y^X is completely independent of y^A and y^H.

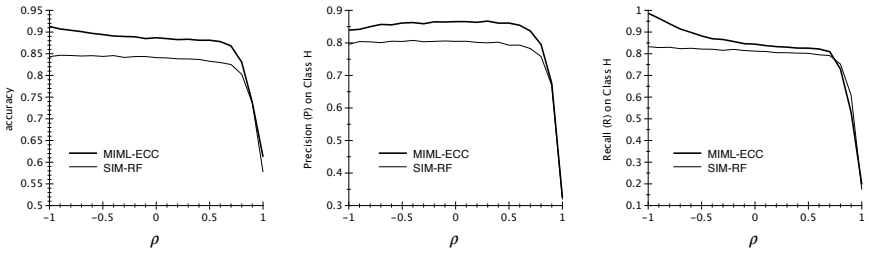
We evaluate the predictions in these experiments based on three measures as functions of ρ: accuracy over all classes, precision on H, and recall on H. Precision and recall are only considered for class H because it plays a central role in this experiment, as it is correlated with A, and easily confused with X. In the second variant of the experiment, noise instances are skipped for the purposes of computing these statistics, because it is not possible for the classifier to predict the correct label (as in [4]). Precision P and recall R are computed as follows:

$$P = \frac{\sum I[y = \hat{y} = H]}{\sum I[\hat{y} = H]},\ R = \frac{\sum I[y = \hat{y} = H]}{\sum I[y = H]} \tag{13}$$

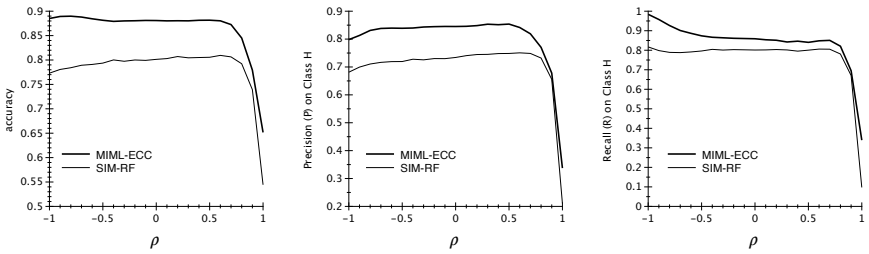
where y is the true label for an instance, and $\hat{y}$ is the predicted label.

Controlled Correlation Results Figures 5a–c and 5d–f show the results of the correlation experiments without and with noise, respectively. We highlight several conclusions based on these results.

First, MIML-ECC generally achieves better accuracy on all classes, and better precision and recall on class H, than SIM-RF across the full range of values for ρ, despite both methods using the same total number of decision trees. Toward the extremes of $\rho = -1$ or $\rho = +1$, this result may be attributed to MIML-ECC's ability to exploit label correlation. Interestingly, MIML-ECC still outperforms SIM-RF, even at $\rho = 0$ or $\hat{\rho} = 0$. This result may be explained by MIML-ECC producing a more diverse ensemble, e.g., because there is more variety in the support instances it selects for training. In the case where there is no noise and



(a) Accuracy (without noise) (b) Precision (without noise) (c) Recall (without noise)



(d) Accuracy (with noise) (e) Precision (with noise) (f) Recall (with noise)

Fig. 5. Controlled correlation results. In a–c, there is no noise class, and bags with empty label sets are rejected. In d–f, there is a noise class, and empty label sets are allowed.

empty label sets are rejected, knowing any two bits of the label set provides information about the third, hence even if the pairwise correlation is 0, there is still some higher order correlation that MIML-ECC can exploit. In the case where there is noise and empty label sets are allowed, MIML-ECC may still have an advantage when $\rho = 0$ because knowing that one class is present implies not all instances can belong to the other classes. For example, if the chain order is A, H, X, and MIML-ECC predicts A is present, subsequent predictions for the probability of an instance to belong to H or X might be reduced. It is also notable that the gap between MIML-ECC and SIM-RF is more pronounced in the experiment with noise, which suggests that MIML-ECC is more robust to noise.

As ρ approaches 1, both classifiers have lower performance according to all three performance measures. The explanation for this result is simple: when $\rho = 1$, there are no unambiguous examples of A or H, because they always occur together, or not at all. Hence there is a fundamental limit to the accuracy any classifier can achieve in this case. However, there is a slight positive trend in the recall for MIML-ECC in Fig. 5f, which suggests that MIML-ECC is gaining some benefit from positive correlation, although this effect is overpowered by the lack of unambiguous training examples.

As ρ approaches -1, the accuracy and recall for MIML-ECC increases, whereas SIM-RF remains *comparatively* flat across the range of values for ρ. One effect that occurs as ρ approaches -1 is that there are more unambiguous examples of A and H, although there are no unambiguous examples of X (because either A or H is always present). However, the widening gap between MIML-ECC and SIM-

RF as ρ approaches -1 indicates that the improved performance of MIML-ECC cannot be attributed entirely to training with fewer unambiguous examples, and is instead caused by MIML-ECC exploiting negative label correlation.

Another trend in the results is visible in Figs. 5d and 5e: the accuracy and precision for SIM-RF slope slightly downward as ρ goes from 1 to -1. In particular, as ρ approaches -1, there is a clear dip in both overall accuracy, and the precision on class H. This can be explained by the fact that as ρ decreases and approaches -1, it becomes less likely to see a training bag that contains only X. Note that when $\rho = -1$, X never appears alone in a bag. Since X is already easily confused with H, this makes it progressively more difficult to correctly classify X, decreasing the overall accuracy as well as the precision on H.

In conclusion, we see that MIML-ECC benefits more from correlation than SIM-RF, particularly when correlation is negative, and is also more robust to noise instances. However, these effects are intertwined with varying levels of ambiguity.

6. Related Work

Graphical models for MIML sometimes include instance labels as hidden variables. Inference over these hidden variables can be used for instance annotation. In addition to LSB-CMM, some recent examples of graphical models for MIML include Dirichlet-Bernoulli Alignment [33] and Exponential Multinomial Mixture model [32]. [36] proposed MLMIL, a conditional random field for MIML which uses Gibbs sampling to infer instance labels.

[28] developed a MIML SVM algorithm which uses a bag-level kernel. Their algorithm predicts instance labels by applying the bag-level classifier to a bag of one instance. [27] proposed a MIML instance annotation algorithm which alternates between sampling random instance labels and training a Semantic Texton Forest (a specialization of RF to images). [22] proposed a MIML algorithm which alternates between assigning instance labels and training a maximum margin classifier. [17] considers the problem of selecting a set of instances explaining each label, which is different from instance annotation, where the goal is to label all instances.

Several formulations besides the max model have been used for MIL and MIML to relate instance and bag labels. Different formulations encode different assumptions about instance labels. One version of the Diverse Density algorithm for MIL [21] used a Noisy-OR model $P(y = 1|B, \theta) = 1 - \prod_{\mathbf{x}_i \in B} \Big(1 - P(y_i = 1|\mathbf{x}, \theta)\Big)$. [20] points out that the max model makes fewer independence assumptions than the Noisy-OR model, although both generate similar probabilities in many cases. In later work the EM-DD [38] algorithm replaced Noisy-OR with max. [24] proposed Multiple-Instance Logistic Regression, which uses a smooth softmax approximation to max. [3, 4] used a multi-class softmax model. [30] propose a model where the bag-label probability is the average of the instance-label probabilities.

Prior work on context-aware learning considered multi-instance learning problems where instances in the same bag are inter-related (non-i.i.d) [16, 15, 40]. A common theme of these approaches is to use graphs to encode inter-instance relationships (e.g., spatial relationship) within a bag, which is then used to help

make more accurate bag-level predictions. This line of work differs from ours in two ways. First, they are primarily interested in improving bag-level predictions by considering the structure within bags. In contrast, we are interested in instance-level predictions. Second, these approaches model context as relationship among instances, whereas our work focuses on context provided by the bag label set. One possible direction for future work is to investigate how both types of context can be used to help with instance annotation.

7. Conclusion & Future Work

We proposed MIML-ECC, an algorithm for context-aware MIML instance annotation. Experiments on image, audio, and artificial datasets show that MIML-ECC achieves better accuracy than other recent algorithms. MIML-ECC is asymptotically efficient, and does not require parameter tuning. Further experiments provide runtime results, suggest that the improved accuracy cannot be attributed only to switching from an SVM-style base classifier to a RF, that ensemble is beneficial, and that MIML-ECC's improved accuracy is related to exploiting correlation in the bag label sets.

MIML-ECC exploits context through correlations, which can be summarized by statements like "if A is present, B is also likely to be present." However, MIML-ECC cannot exploit a different kind of context, which can be summarized as "if one A is present, there are likely to be more A's." For example, consider Fig 1b. It might be easy to recognize some of the larger cows in the image, but harder to recognize the small ones. However, after recognizing one cow, it we might expect to find more cows. MIML-ECC will not exploit this kind of context because it can only use information about the presence or absence of other classes to inform its prediction. A useful direction for future work is to develop algorithms for MIML instance annotation that can exploit both bag-level label correlations, and relationships between instances, as in [16, 15, 40]. Similar correlation structures have been exploited in MLC using a collective classification / relational learning approach [14].

We made several assumptions in formulating MIML-ECC; it is interesting to explore related models with different assumptions. For example, we assumed each instance has exactly one label. However, there are cases were instances have none of the labels in the set of known classes (e.g., clutter in an image), and also where an instance should have multiple labels.

Acknowledgements. This work is partially funded by NSF grant 1055113 to Xiaoli Z. Fern, and the College of Engineering, Oregon State University.

References

[1] S. Andrews, I. Tsochantaridis, and T. Hofmann. Support vector machines for multiple-instance learning. *Advances in Neural Information Processing Systems*, 15:561–568, 2002.

[2] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[3] F. Briggs, X. Fern, and R. Raich. Rank-loss support instance machines for miml instance annotation. In *International Conference on Data Mining*, pages 534–542, 2012.

[4] F. Briggs, X. Fern, R. Raich, and Q. Lou. Instance annotation for multi-instance multi-label learning. *Transactions on Knowledge Discovery from Data (TKDD), 2012*, 2012.

[5] F. Briggs, B. Lakshminarayanan, L. Neal, X. Fern, R. Raich, S. Hadley, A. Hadley, and M. Betts. Acoustic classification of multiple simultaneous bird species: A multi-instance multi-label approach. *Journal of the Acoustical Society of America*, 131:4640, 2012.

[6] A. Buja, W. Stuetzle, and Y. Shen. Loss functions for binary class probability estimation and classification: Structure and applications. Technical report, 2005.

[7] L. Carroll. *Through the looking-glass: and what Alice found there*. 1896.

[8] T. Cour, B. Sapp, and B. Taskar. Learning from partial labels. *Journal of Machine Learning Research*, 12:1225–1261, 2011.

[9] K. Dembczynski, W. Cheng, and E. Hüllermeier. Bayes optimal multilabel classification via probabilistic classifier chains. In *International Conference on Machine Learning*, pages 279–286, 2010.

[10] J. Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30, 2006.

[11] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, June 2010.

[12] P. W. Frey and D. J. Slate. Letter recognition using holland-style adaptive classifiers. *Machine Learning*, 6:161, 1991.

[13] R. Frost. *Mountain Interval*. 1916.

[14] X. Kong, X. Shi, and S. Y. Philip. Multi-label collective classification. In *SDM*, volume 11, pages 618–629, 2011.

[15] B. Li, W. Xiong, and W. Hu. Context-aware multi-instance learning based on hierarchical sparse representation. In *Data Mining (ICDM), 2011 IEEE 11th International Conference on*, pages 370–377, 2011.

[16] B. Li, W. Xiong, and W. Hu. Web horror image recognition based on context-aware multi-instance learning. In *International Conference on Data Mining*, pages 1158–1163, 2011.

[17] Y. Li, J. Hu, Y. Jiang, and Z. Zhou. Towards discovering what patterns trigger what labels. In *Conference on Artificial Intelligence*, 2012.

[18] Y. Li, S. Ji, S. Kumar, J. Ye, Z. Zhou, et al. Drosophila gene expression pattern annotation through multi-instance multi-label learning. In *International Joint Conference on Artificial Intelligence*, pages 1445–1450, 2009.

[19] L. Liu and T. Dietterich. A conditional multinomial mixture model for superset label learning. In *Advances in Neural Information Processing Systems*, pages 557–565, 2012.

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

[21] O. Maron and T. Lozano-Pérez. A framework for multiple-instance learning. *Advances in Neural Information Processing Systems*, pages 570–576, 1998.

[22] N. Nguyen. A new svm approach to multi-instance multi-label learning. In *International Conference on Data Mining*, pages 384–392, 2010.

[23] A. Rahimi and B. Recht. Random features for large-scale kernel machines. *Advances in Neural Information Processing Systems*, 20:1177–1184, 2007.

[24] S. Ray and M. Craven. Supervised versus multiple instance learning: An empirical comparison. In *International Conference on Machine Learning*, pages 697–704. ACM, 2005.

[25] J. Read, B. Pfahringer, G. Holmes, and E. Frank. Classifier chains for multi-label classification. *Machine Learning*, 85(3):333–359, 2011.

[26] C. Shen, J. Jiao, B. Wang, and Y. Yang. Multi-Instance Multi-Label Learning For Automatic Tag Recommendation. In *Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics (SMC 2009)*, 2009.

[27] A. Vezhnevets, J. Buhmann, and E. Zurich. Towards Weakly Supervised Semantic Segmentation by Means of Multiple Instance and Multitask Learning. In *Conference on Computer Vision and Pattern Recognition*, 2010.

[28] S. Vijayanarasimhan and K. Grauman. What's it going to cost you?: Predicting effort vs. informativeness for multi-label image annotations. In *Conference on Computer Vision and Pattern Recognition*, pages 2262–2269, 2009.

[29] J. Winn, A. Criminisi, and T. Minka. Object categorization by learned universal visual dictionary. In *International Conference on Computer Vision*, pages 1800–1807, 2005.

[30] X. Xu and E. Frank. Logistic regression and boosting for labeled bags of instances. *Advances in Knowledge Discovery and Data Mining*, pages 272–281, 2004.

[31] X. Xu, X. Xue, and Z. Zhou. Ensemble multi-instance multi-label learning approach for video annotation task. In *Proceedings of the 19th ACM international conference on Multimedia*, pages 1153–1156. ACM, 2011.

[32] S. Yang, J. Bian, and H. Zha. Hybrid Generative/Discriminative Learning for Automatic Image Annotation. In *Conference on Uncertainty in Artificial Intelligence*, 2010.

[33] S. Yang, H. Zha, and B. Hu. Dirichlet-bernoulli alignment: A generative model for multi-class multi-label multi-instance corpora. In *Advances in Neural Information Processing Systems*, pages 2143–2150, 2009.

[34] W. Yang, Y. Wang, A. Vahdat, and G. Mori. Kernel latent svm for visual recognition. In *Advances in Neural Information Processing Systems*, volume 2, page 4, 2012.

[35] A. Yuille and A. Rangarajan. The concave-convex procedure (cccp). *Advances in Neural Information Processing Systems*, 2:1033–1040, 2002.

[36] Z. Zha, X. Hua, T. Mei, J. Wang, G. Qi, and Z. Wang. Joint multi-label multi-instance learning for image classification. In *Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[37] M. Zhang and Z. Zhou. M3MIML: A maximum margin method for multi-instance multi-label learning. In *International Conference on Data Mining*, pages 688–697, 2008.

[38] Q. Zhang and S. Goldman. EM-DD: An improved multiple-instance learning technique. *Advances in Neural Information Processing Systems*, 2:1073–1080, 2002.

[39] Z. Zhou and M. Zhang. Multi-instance multi-label learning with application to scene classification. *Advances in Neural Information Processing Systems*, 19:1609, 2007.

[40] Z.-H. Zhou, Y.-Y. Sun, and Y.-F. Li. Multi-instance learning by treating instances as non-iid samples. In *International Conference on Machine Learning*, pages 1249–1256, 2009.

[41] Z.-H. Zhou, M.-L. Zhang, S.-J. Huang, and Y.-F. Li. Multi-instance multi-label learning. *Artificial Intelligence*, 176(1):2291–2320, 2012.

Appendix 1

The correlation coefficient $\rho(X, Y)$ between two random variables X and Y is given by

$$\rho(X,Y) = \frac{\mathrm{Cov}(X,Y)}{\sqrt{\mathrm{Var}(X)}\sqrt{\mathrm{Var}(Y)}} \tag{14}$$

Let X be a Bernoulli RV with $P(X = 1) = \frac{1}{2}$. Similarly, let Y conditioned on X be a Bernoulli RV with $P(Y = 1|X) = \frac{1}{2}(1 + \rho)X + \frac{1}{2}(1 - \rho)(1 - X)$, as in Sec. 5.5, eqn. (12). The correlation coefficient $\rho(X, Y) = \rho$.

Proof: we begin by noting the property that the expected value of an arbitrary Bernoulli RV T with $P(T = 1) = p$ satisfies $E[T] = P(T = 1) = p$. Moreover, since $T \in \{0, 1\}$, $T^k = T$ for $k = 1, 2, \ldots$ and consequently $E[T^k] = p$. The variance of T is given by $\mathrm{Var}(T) = E[T^2] - E[T]^2 = p - p^2 = p(1-p)$. To compute the correlation coefficient $\rho(X, Y)$, we first compute $E[X]$, $E[Y]$, $\mathrm{Var}(X)$, $\mathrm{Var}(Y)$, and $\mathrm{Cov}(X, Y)$. Since $P(X = 1) = \frac{1}{2}$, we have $E[X] = P(X = 1) = \frac{1}{2}$. The expectation of Y is computed as follows

$$\begin{aligned}
E[Y] &= E_X[E_Y[Y|X]] \\
&= E_X[P(Y = 1|X)] \\
&= E_X[\frac{1}{2}(1 + \rho)X + \frac{1}{2}(1 - \rho)(1 - X)] \\
&= \frac{1}{2}(1 + \rho)E_X[X] + \frac{1}{2}(1 - \rho)(1 - E[X]) \\
&= \frac{1}{2}(1 + \rho)\frac{1}{2} + \frac{1}{2}(1 - \rho)\frac{1}{2} \\
&= \frac{1}{2}.
\end{aligned} \tag{15}$$

Since X and Y are Bernoulli RVs with $P(X = 1) = P(Y = 1) = E[X] = E[Y] = \frac{1}{2}$, we also have $\mathrm{Var}(X) = \mathrm{Var}(Y) = \frac{1}{2}(1 - \frac{1}{2}) = \frac{1}{4}$. Next, we compute

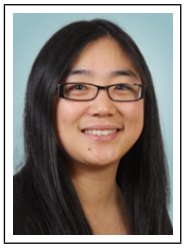
$$\begin{aligned}
E[XY] &= E_X[E_Y[XY|X]] \\
&= E_X[XE_Y[Y|X]] \\
&= E_X[XP(Y = 1|X)] \\
&= E_X[X(\frac{1}{2}(1 + \rho)X + \frac{1}{2}(1 - \rho)(1 - X))]
\end{aligned}$$

$$= \frac{1}{2}(1+\rho)E_X[X^2] + \frac{1}{2}(1-\rho)E[X(1-X)]$$
$$= \frac{1}{2}(1+\rho)\frac{1}{2}$$
$$= \frac{1}{4}(1+\rho). \quad (16)$$

The covariance is therefore $\mathrm{Cov}(X,Y) = E[XY] - E[X]E[Y] = \frac{1}{4}(1+\rho) - \frac{1}{2}^2 = \frac{1}{4}\rho$. Finally, substituting $\mathrm{Var}(X) = \mathrm{Var}(Y) = \frac{1}{4}$ and $\mathrm{Cov}(X,Y) = \frac{1}{4}\rho$ into (14), yields $\rho(X,Y) = \rho$.

Author Biographies

Forrest Briggs received his B.S. from Harvey Mudd College in 2006, and Ph.D. from Oregon State University in 2013, both in Computer Science. His work at OSU was partially funded by an NSF IGERT fellowship for Ecosystems Informatics, and he also received a minor in Ecosystems Informatics. From 2006 to 2008, he was a software developer at Sticky, Inc., and The Learning Annex LLC. As of 2014, he is a machine learning scientist at Rocket Fuel Inc. His research interests include multi-instance and multi-label supervised classification, bioacoustics, and big data.



Xiaoli Z. Fern is an associate professor at the School of Electrical Engineering and Computer Science, Oregon State University, Corvallis, OR. She received her Ph.D. degree in Computer Engineering from Purdue University and her M.S. and B.S. degrees from Shanghai Jiao Tong University. Her general research interests are in the areas of machine learning and data mining. She received an NSF Career Award in 2011. Dr. Xiaoli Fern is on the editorial board of the Machine Learning Journal and serves regularly on the program committee for a number of top tier international conferences on machine learning and data mining such as ICML, ECML, AAAI, KDD, ICDM, and SIAM SDM.

Raviv Raich received B.Sc. and M.Sc. degrees from Tel Aviv University, Tel-Aviv, Israel, in 1994 and 1998, respectively, and a Ph.D. degree from the Georgia Institute of Technology, Atlanta, in 2004, all in electrical engineering. Between 1999 and 2000, he was a Researcher with the Communications Team, Industrial Research, Ltd., Wellington, New Zealand. From 2004 to 2007, he was a Postdoctoral Fellow with the University of Michigan, Ann Arbor. From 2007 to 2013, he was an Assistant Professor at the School of Electrical Engineering and Computer Science, Oregon State University, Corvallis. Since fall 2013, he is an Associate Professor at the school of Electrical Engineering and Computer Science, Oregon State University, Corvallis. His research interests are in statistical signal processing and machine learning. He has particular interest in applications concerning structure discovery in high dimensions. Dr. Raich serves as an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING. He is a member of the Machine Learning for Signal Processing (MLSP) Technical Committee of the IEEE Signal Processing Society.

Correspondence and offprint requests to: Forrest Briggs, School of Electrical Engineering of Computer Science, Oregon State University, Corvallis, OR 97331-5501, USA. Email: fbriggs@gmail.com