Multi-Instance Multi-Label Learning for Image Classification with Large Vocabularies

Oksana Yakhnenko
oksana@google.com
Google Inc
New York, NY, USA

Vasant Honavar
honavar@cs.iastate.edu
Iowa State University
Ames, IA, USA

Abstract

Multiple Instance Multiple Label learning problem has received much attention in machine learning and computer vision literature due to its applications in image classification and object detection. However, the current state-of-the-art solutions to this problem lack scalability and cannot be applied to datasets with a large number of instances and a large number of labels. In this paper we present a novel learning algorithm for Multiple Instance Multiple Label learning that is scalable for large datasets and performs comparable to the state-of-the-art algorithms. The proposed algorithm trains a set of discriminative multiple instance classifiers (one for each label in the vocabulary of all possible labels) and models the correlations among labels by finding a low rank weight matrix thus forcing the classifiers to share weights. This algorithm is a linear model unlike the state-of-the-art kernel methods which need to compute the kernel matrix. The model parameters are efficiently learned by solving an unconstrained optimization problem for which Stochastic Gradient Descent can be used to avoid storing all the data in memory.

1 Introduction

Image classification is a challenging task with many applications in computer vision, including image auto-annotation and content-based image retrieval. Images are naturally encoded as high dimensional vectors of pixels and the extraction of meaningful features from the images to this day is a difficult problem. In addition, images are annotated with multiple keywords that may or may not be correlated. Recent state-of-the-art image classification and annotation approaches [19, 20] used global features extracted from the images. However, the global features may not be well-suited in when images contain multiple objects. The keywords assigned to an image often correspond to the individual regions (or objects) present in the images as well as the collection of the objects. Therefore, global features may not necessary capture the region-level information present in the images. On the other hand, image segmentation algorithms (such as normalized cuts [25]) attempt to discover regions in the image that may correspond to objects, and hence they make it possible to represent images as a collection of regions (objects). Therefore, image classification may be naturally modeled as a Multiple Instance Multiple Label (MIML) learning problem [30]. MIML learning

BMVC 2011 http://dx.doi.org/10.5244/C.25.59

This work was supported in part by the grant IIS 0711356 from the National Science Foundation and by the Iowa State University Center for Computational Intelligence, Learning, and Discovery.

is a generalization of traditional supervised learning. The input data is no longer a set of labeled instances, but a set of labeled bags (that contain multiple instances), each associated with a set of labels. Current state-of-the-art approaches to MIML fall into two categories: probabilistic models (i.e. [30]) and kernel methods (i.e. [28, 32]). Inference problem for the probabilistic approach is intractable and cannot be solved exactly. Therefore the solution is usually approximated using Loopy Belief propagation approach, or Gibbs Sampling. This poses two problems: the likelihood is inexact, and the inference is slow. It was shown that kernel methods achieve better performance, however in order to learn the kernel model the bag-kernel needs to be computed [28]. Therefore, the run-time of the algorithm is at least N^2 where N is the number of all possible instances. In addition, the predictive power of approach in [28] relies on the use of one-against-one SVMs.

Overview and Contributions In this paper we introduce an algorithm that is scalable for tasks where the number of bags and the number of instances can be large.[1] In order to do so, we focus on a *linear* model, parameters for which can be learned by solving an optimization problem in the primal. In particular, in supervised learning the state-of-the art scalable algorithms are linear models solved in primal, as for example SVM primal solver [5], PEGASOS [24] and LIBLINEAR [11]. In particular, recent work in explicity embedding the features into some other space that allows to compute the kernel using its dot product defition [13] has successfully employed linear SVM solvers in order to avoid computing the kernel matrix. We model our solution to the MIML learning problem within the well studied loss-penalty formulation that allows the algorithm to trade-off the loss (classification error) against the regularization term (penalty for the classifiers complexity in order to avoid overfitting on the training data). Our loss function is specifically designed for Multiple Instance learning. We model the loss of a bag as the negative logarithm of the probability of the label given the bag. The probability of a label given the bag is computed using logistic Noisy-Or model [8, 22, 29]. We consider several regularization functions, such as well-studied Frobenius-norm and ℓ_1-norm regularization terms. In addition, we use Trace Norm penalty as the regularization term in order to model and to capture the correlations among labels associated with an image. Trace Norm regularization has previously been used to achieve a similar goal in the case of Multiple Label learning [2], multi-task learning [3] and collaborative filtering [1]. The results of our experiments on the Microsoft dataset show that the proposed MIML method outperforms the Multiple Instance kernel algorithm proposed by [28] and are competitive with the MIML classifiers proposed by [30]. Our experiments with other large-scale datasets with a large number of labels show that the proposed approach scales well to settings in which it is not feasible to apply existing MIML learning algorithms [28, 30].

2 Related Work

We begin with describing the related work in Multiple Instance Multiple Label learning as well as summarize the recent state-of-the-art algorithms for this problem. Zhou and Zhang [31] proposed two solutions to MIML. The first approach involves effectively transforming each bag of instances into a single instance and then applying a single instance Multiple Label learning algorithm to the resulting dataset. The second approach generalizes a Multiple

[1] Implementation is available at http://www.cs.iastate.edu/~oksayakh/research/resources.html

Instance single label learning algorithm to handle Multiple Labels yielding a MIML learning algorithm. Zha et al. [30] proposed a discriminative model based on a collective multi-label model [15] which was shown to yield state-of-the-art results on several image classification tasks that are naturally posed as MIML learning problems. They proposed an undirected graphical model, and in order to model a valid probability of labels given the bags, it is necessary to compute the summation over all possible label assignments for each bag. The computation of the probability is therefore exponential in the number of possible labels, and the exact solution is intractable. Zha et al. [30] proposed an approximation for this computation using Gibbs sampling. Gibbs sampling, however, can be slow when the number of labels is large, making the algorithm impractical for large datasets with a large number of possible labels. In particular, the Joint MIML model [30] solves an optimization problem. Each step uses Gibbs Sampling to estimate the probability of the labels given a bag and thus each step needs $O(DNK)$ (N is the number of instances, D is the number of features and K is the number of Gibbs Sampler iterations, typically very large to ensure accurate estimates). Vijayanarasimhan and Grauman [28] proposed another MIML label algorithm which relies on a Multiple Instance kernel [14] and one-versus-one Support Vector Machine training. Therefore, in order to train the model it is required to compute the kernel (at least $O(N^2)$ where N is the number of instances) and it is required to train M^2 SVMs where M is the number of possible labels). As with previous formulations, this approach is only applicable for small number of labels.

3 Preliminaries

We briefly review Multiple Label learning, Multiple Instance learning and Multiple Instance Multiple Label learning.

Single Instance Single Label Learning Let $\mathscr{R}^d$ be a d-dimensional vector space and let $\mathscr{L} = \{l_1, ..., l_M\}$ be a set of labels. Given the dataset $D = \{x_i, y_i\}$ where $x_i \in \mathscr{R}^d$ and $y_i \in \mathscr{L}$ the goal of supervised learning is to learn a function $f : \mathscr{R}^d \rightarrow \mathscr{L}$.

Multiple Label Learning The first generalization of supervised learning is multiple label learning. Each observation x_i has a collection of labels assigned to it: $Y_i = \left\{y_i^1, ..., y_i^{m_i}\right\}$ where m_i is the number of labels. Each of the labels is drawn from the label space $y_i^j \in \mathscr{L} = \{l_1, ..., l_M\}$. If the number of labels is 1, this scenario reduces to a standard single-label classification task.

Multiple Instance Learning The second generalization of single instance learning is the Multiple Instance learning [8]. Multiple Instance learning scenario assumes that the label y_i is assigned to a bag of instances $X_i = \left\{x_{i1}, ..., x_{ik_i}\right\}$ where k_i is the number of instances in the bag and each instance $x_{ij} \in R^d$ where d is the dimensionality of the feature space used to represents each instance. The bag is assigned a positive label $y_i = 1$ if at least one of the instances in X_i is positive (however it is not known which one is positive). The bag is assigned a negative label $y_i = -1$ if it only contains only negative instances.

Multiple Instance Multiple Label Learning Multiple Instance Multiple Label learning [31] is a natural generalization of the Multiple Instance learning. It takes as input pairs $\{X_i, Y_i\}_{i=1}^N$ where each $X_i = \left\{x_{i1}, ..., x_{ik_1}\right\}$ is a bag of instances labeled with a set of labels $Y_i = \{y_i^1, ..., y_i^{m_i}\}$ where each of the labels y_j is drawn from a set of possible labels $\mathscr{L} = \{l_1, ..., l_M\}$. If the size of the training bags is one and it is labeled with only one label, the learning task reduces to the traditional supervised learning.

4 Discriminative Multiple Instance Multiple Label Learning

We now proceed to introducing our solution to the MIML learning problem. Our approach is to adapt a well-studied framework for learning classifiers. In this framework, a classification function is learned in order to minimize the trade-off between the classifier loss (classification error) and the complexity of the classifier. We begin with an observation that the Log Loss for single instance learning is related to the probability of predicting the label correctly, and we generalize the loss to the Multiple Instance setting by using a Noisy-Or model [8, 29] to compute the probability of the label that the bag can take given the instances in the bag. We also consider several well-known penalty functions, including Trace Norm, Frobenius Norm and ℓ_1-norm [2], which to the best of our knowledge have not been considered for MIML. Our formulation of a learning task is a *linear* model that can be efficiently learned in the primal without the need to compute the kernel or to solve an expensive quadratic programming problem.

4.1 Discriminative MIML Learning

The general formulation of learning [27] suggests learning a classifier by trading off between the classifier's average empirical loss and the complexity of the classifier. This reduces to choosing, from a class of functions H, a function h^* that minimizes a weighted combination of the loss and the penalty:

$$h^* = \min_h \left(\sum_{i=1}^{N} \text{loss}(y_i, h(x_i)) + C\text{penalty}\,(h) \right)$$

where C is a constant that controls the amount of trade-off. This formulation has been extended to multiple label learning [7] by training a collection of classifiers, each parametrized by a weight vector w_j for each class l_j by decomposing the loss over each label for each instance. Let there be M classifiers $h_1 ... h_M$ (one for each of the M classes, or equivalently, classifiers $h_1 ... h_M$ predicting the corresponding elements of the vector of binary labels $y_i^1, y_i^2 .... y_i^M$, so that $y_i^j = 1$ if l_j is a label assigned to x_i and $y_i^j = -1$ otherwise).

$$\{h_1 ... h_M\}^* = \min_{h_1 ... h_M} \sum_{i=1}^{N} \sum_{j=1}^{M} \text{loss}\left(y_i^j, h_j(x_i)\right) + C\text{penalty}\,(h_1, ..., h_M)$$

We adapt this framework to Multiple Instance Multiple Label learning by 1) designing a loss function that models the loss for the *bag of instances* and 2) choosing a penalty function that is specifically suitable for Multiple Label learning.

4.2 Loss Function for Multiple Instance Learning

We begin with discussion of loss functions for single instance learning and then propose the loss function appropriate for Multiple Instance learning that computes the loss using bag of instances directly. Among the studied loss functions are Hinge Loss, Logistic loss and Squared-Loss. It is well-known that using Hinge Loss (defined as $h(z) = \max(0, 1-z)$)

[2] By a slight abuse of notation we will use ℓ_p-norm definition for both, vectors, and matrices: $||w||_p = (\sum_i w_i^p)^{\frac{1}{p}}$

and ℓ_2-norm penalty on the classifier weight vector w results in an SVM model. The main complication with using Hinge Loss is that it is not differentiable. Therefore Hinge Loss is frequently approximated with some differentiable function. For example, [2] used a generalized Log Loss [3] and [19] used a numerical approximation of the Hinge Loss. Logistic Loss, on the other hand, is differentiable. It can be viewed as a smooth approximation to the Hinge Loss function. In addition, it was proved [23] that Logistic Loss also results in maximum margin classifiers [23]. Logistic Loss can also be interpreted as a probability of assigning the correct class to an input. Now we turn to the problem of modeling the loss for a bag of instances. How can we compute the loss $l(y_i^j, h_j(x_i))$ for bags of instances when the labels for individual instances in a bag are unknown? Consider the Log Loss:

$$l(y_i^j, h_j(x_i)) = -\log p(y_i^j|x_i) = -\log \frac{1}{1+\exp(-y_i^j h_j(x_i))}$$

In the case of logistic regression, Log Loss is the negative log of the probability of y_i^j given the observation x_i. Hence,

$$l(y_i^j, h_j(x_i)) = -\delta\left(y_i^j, 1\right)\log p(y_i^j = 1|x_i) - \delta\left(y_i^j, -1\right)\log p(y_i^j = -1|x_i)$$

where $\delta(a,b) = 1$ if $a = b$ and 0 otherwise. Let y_{ik}^j denote the jth bit of the vector of class labels for the kth *instance* x_{ik} in the ith bag x_i. Let A^T be the transpose of a matrix A. If we are given a classifier defined by w_j with respect to membership in class l_j, we can use sigmoid function to model the probability that the kth *instance* x_{ik} in the ith bag x_i is positive (with respect to membership in class label l_j): $p(y_{ik}^j = 1|x_{ik}) = \sigma(w_j^T x_{ik}) = \frac{1}{1+\exp(-w_j^T x_{ij})}$. Then the probability that the instance is negative with respect to membership in the jth class is given by $1 - p(y_{ik}^j = 1|x_{ik})$. Because a bag is labeled negative only if all the instances in it are negative, we can use a Noisy-Or model to combine the probabilities that the individual instances in the bag are negative: $p(y_i^j = -1|x_i, w_j) = \prod_{k=1}^{K_i}\left(1 - p(y_i^j|x_{ik}, w_j)\right) = \prod_{k=1}^{K_i}\left(1 - \sigma(w_j^T x_{ik})\right)$ The probability that the bag is positive is then given by $p(y_i^j = 1|x_i, w_j) = 1 - p(y_i^j = -1|x_i, w_j)$ and therefore we have all the pieces necessary to compute the loss over a bag.

4.3 Penalty functions for correlated Multiple Labels

The choice of an appropriate penalty function has been an active research area. Traditionally, ℓ_2-norm has been used as it it closely related with the definition of margin [27], however ℓ_1-norm has also been extensively used, and was shown to result in sparse classifier weights. However, using either ℓ_1 or ℓ_2 norm penalties in the multi-label setting is equivalent to training M one-against-all independent classifiers. This is not desirable when the labels are correlated. Recently, Trace-Norm penalty has been proposed in the setting for multiple label learning when the labels are correlated [2, 3, 19] and it was shown to capture the correlations among labels unlike other norms. Let $W = [w_1, ..., w_M]$ be a matrix of weights that correspond to multiple instance classifiers where w_j is a vector that defines a multiple instance classifier for class j. The Trace Norm $\|W\|_\Sigma$ is defined as

$$\min_{W=FG} \frac{1}{2}\left(\|F\|_{\mathscr{F}}^2 + \|G\|_{\mathscr{F}}^2\right)$$

[3] Defined as $g(z,\gamma) = \frac{1}{\gamma}\log(1+\exp(\gamma(1-z)))$ and it approximates Hinge Loss as $\gamma \to \infty$.

where $\|\cdot\|_{\mathscr{F}}$ is the Frobenius norm (another name for the matrix ℓ_2 norm). Trace Norm factorizes classifier weights matrix W into the matrices F and G, where F maps the inputs to some feature space and G performs classification in that space. However this factorization is not needed explicitly in order to compute the norm. It was shown [2] that the penalty term $\|W\|_\Sigma$ is equivalent to the sum of absolute values of the singular values of the matrix: $\|W\|_\Sigma = \sum|\gamma_i|$ where γ is a vector of singular values of W and $|\cdot|$ is the absolute value and therefore only the SVD of W needs to be computed. The existence of trace-norm SVM [2] that has been applied to Multiple Label learning problems in computer vision [19] leads us to proposing another simple solution to MIML similar in spirit to MIML-SVM [31]. Instead of learning independent binary SVMs, we propose learning correlated multiple-class classifiers [2] (using hinge loss and trace norm as described above). The transformation from Multiple Instances to single instances is described in detail in [31]. We will refer to this learning algorithm as **MI-MatFact**.

4.4 Solving the Optimization Problem

The model parameters W can be learned by solving an unconstrained optimization problem. The goal is to find weight matrix W^* that minimizes $J = J_{loss} + J_{reg}$, where $J_{loss} = \sum_{i=1}^{N}\sum_{j=1}^{M} loss(y_i^j, h_j(x_i))$ and $J_{reg} = C\|W\|_\Sigma$. This is an unconstrained minimization problem, and therefore it can be solved using any unconstrained minimization method [21]. We use Limited Memory BFGS [18] since it is known to converge to the solution faster without the need to explicitly compute and store the Hessian (which can be expensive or infeasible for the large number of model parameters). It is well known that the Noisy-Or function is not a convex function and therefore its use makes the objective function not convex. The solution may be a local optimal solution.[4]

In order to optimize the objective function, one needs to compute the gradient of the objective function with respect to the model parameters. The gradient of the loss function with respect to the jth column of the weight matrix W is

$$\frac{\partial J_{loss}}{\partial w_j} = -\sum_{i=1}^{N}\left(\delta\left(y_i^j,1\right)\frac{\left(1-p(y_i^j=1|x_i)\right)}{p(y_i^j=1|x_i)} - \delta\left(y_i^j,-1\right)\sum_{k=1}^{K_i}\sigma(w_j^T x_{ik})x_{ik}\right)$$

The computation of the gradient of the regularization term requires more work as the absolute value function is not differentiable at 0. It however can be approximated with a differentiable everywhere smooth function $a(\cdot)$ and therefore the Trace Norm regularization term becomes $\|W\|_\Sigma = \sum a_\tau(\gamma_i)$ [2] where

$$a_\tau(x) = \begin{cases} |x| & |x| > \tau \\ \frac{x^2}{2\tau} + \frac{\tau}{2} & |x| \le \tau \end{cases}$$

and τ is a small positive number (we used $\tau = 10^{-9}$) . The gradient of the regularization term is given by:

$$\frac{\partial}{\partial W} J_{reg} = CUa_\tau^{'}(D)V^T$$

[4] While the objective function is not convex, we must not forget that the goal of learning is not convexity of the objective function, but rather a good generalization on the unseen data. In fact non-convexity of the objective function generally does not pose a challenge and non-convex problems were shown to have better performance and be more scalable than convex [6].

where $W = UDV^T$ is the singular value decomposition of W and $a_\tau^{'}(D)$ is a derivative of a_τ applied to each element of diagonal of D. The function $a_\tau(x)$ is twice-differentiable everywhere and its first derivative [5] is

$$a_\tau^{'}(x) = \begin{cases} sign(x) & |x| > \tau \\ \frac{x}{\tau} & |x| \leq \tau \end{cases}$$

We will refer to this learning algorithm as Discriminative Multiple Instance Multiple Label Model with trace-norm regularization (DMIML$_\Sigma$). In addition to Trace Norm regularization we consider regularizing the MIML loss function with $W_1 = \sum_{w_i \in W} |w_i|$ norm (ℓ_1-norm), and $W_2^2 = \sum_{w_i \in W} |w_i|^2$ (ℓ_2-norm) that result in independent classifiers. Training these classifiers is done using a similar optimization problem as above, with the Trace-Norm replaced with ℓ_1 or ℓ_2 respectively. Note that ℓ_1 is defined using sum of the absolute values and therefore it is not differentiable. Therefore, as with Trace Norm, we approximate it using $W_1 = \sum_{w_i \in W} a_\tau(w_i)$. Arguably, it can be costly to compute SVD decomposition of a weight matrix when the number of features and the number of labels is very high (on the order of several thousands). Therefore we also consider ℓ_1 and ℓ_2 regularizations as alternatives, and we refer to these variants of Discriminative Multiple Instance Multiple Label learning as DMIML$_{\ell_1}$ and DMIML$_{\ell_2}$ respectively.

5 Experiments and Results

We now proceed with experimental evaluation of our algorithm and its comparison with other known MIML algorithms including the recent state-of-the-art, namely Joint Multiple Instance Multiple Label approach [30] and one-against-one SVM with Multiple Instance Kernel [28]. We can only compare our approach with [30] and [28] on a relatively small but challenging MSRC dataset. No public implementation is available for [30] and it is non-trivial to implement. [28] requires 1-vs-1 SVM training, which means that for small MSRC dataset 400 SVMs need to be trained. For larger datasets that we use in our evaluation (such as Corel-5k and IAPR TC-12) use of this algorithm is infeasible as these datasets have ~300 labels and thus 90,000 SVMs need to be trained to fully compare with approach in [28] .

5.1 Data

MSRC The version 2 of Microsoft Object Class Recognition dataset consists of 591 images. The dataset also provides pixel level ground truth and each pixel is labeled with one out of 23 possible classes (class 'void' was not used, and following [28, 30] classes 'horse' and 'mountain' were treated as void since they have only a few observations, resulting in 21 classes). This dataset has been used in the past to evaluate MIML classifiers in computer vision [28, 30]. While the dataset provides ground truth at pixel level, this information is not used in training, and only the image and the image-level label information is used. As in [28] we segment the images using normalized cuts [25] and from each region we extract texton features [26] and color histograms. Each segment is then treated as an instance, and each image is treated as a bag of segments.

[5] The correctness of the analytical gradient and its implementation was checked numerically (using finite difference $\nabla f_{num}(x) \approx \frac{f(x+h)-f(x)}{h}$).

Method	MIMLSVM	MIMLBoost [30]	MIMIL [30]	MIL-Kernel [28]	MI-Mat-Fact
Average AUC	0.776 ± 0.02	0.766	0.902	0.896	0.8076 ± 0.02
Method	DMIML$_{\ell_1}$	DMIML$_{\ell_2}$	DMIML$_\Sigma$	DMIML	
Average AUC	0.897±0.011	**0.914±0.014**	0.909 ± 0.013	0.829 ± 0.031	

Table 1: AUC (± standard deviation) for MSRC V2 dataset

IAPR TC-12 One recent benchmark in image annotation and retrieval is the IAPR TC-12 dataset [10]. It consists of 20,000 images each annotated with keywords from 274 categories. Each image has been manually segmented and annotated according to a predefined vocabulary of labels. From each segment the following visual features were extracted: area, boundary/area, width and height of the region, average and standard deviation in x and y, convexity, average, standard deviation and skewness in both color spaces RGB and CIE-Lab. Each segment is treated as an instance, and each image is treated as a bag.

Corel 5K This dataset, first introduced by [9] which is a widely used benchmark for image annotation. The dataset consists of 4500 training images and 500 test images and there are 263 possible keywords. Same features computed from segmented images as used by [9] were used in our experiments.

5.2 Experiments and Results

For each dataset we experiment the following classifiers: **DMIML$_\Sigma$**, **DMIML$_{\ell_1}$**, **DMIML$_{\ell_2}$**, **DMIML** (the discriminative MIML model that uses no regularization with $C = 0$), **MI-MatFact**: (proposed simple solution to MIML by transforming MI to single instance and then applying the multi-class algorithm [2]), **MIML-SVM** (transforming MI to single instance and training binary SVMs [31]). Where applicable, we compare the MIML [30] and MI-kernel [28]. We use cross-validation tune the value of the regularization parameter C for Trace Norm, ℓ_1 and ℓ_2 variants of DMIML. For each dataset we select one value from $\{2^{-9}, 2^{-5}, \ldots 2^5, 2^6\}$ that yields the highest performance on the validation set (subset of training data) then retrain the model for that value of C on the entire training set and evaluate its performance on the test set. We use AUC, the area under ROC curve [16] as the performance measure. Since there are multiple labels in the datasets, we compute the AUC for each label and report the average AUC over all of the possible labels. The AUC enables direct comparisons with the results available in the literature for the other MIML learning methods [28, 30, 31].

MSRC To compare the proposed approach with the current state-of-the-art techniques, following the set-up in [30] we divide the data set into 5 equal parts and repeatedly use one part for testing and the rest for training. In each run, we do 2-fold cross-validation on the training data to pick the optimal value of parameter C. The average over 5 runs for the best performing parameter values was computed. These results are reported in Table 1. The proposed DMIML using ℓ_2 and Trace Norm regularization yield the best performance and both models outperform the Joint MIML model [30] and the one-against-one SVM with MI kernel [28]. Overall, the best performance of the ℓ_2 norm is slightly better than the best performance of the Trace Norm. This is not surprising as this dataset has relatively small number of classes (21) many of which are not correlated.

IAPR TC-12 We split the dataset into 60% (12,000 images) training and 40% testing (8,000 images). The training set was further split into 8,000 training and 4,000 validation images to tune the value of C, after which all images were used to train the final model. The performance of the MIML classifiers is summarized in Table 2.

	MIMLSVM	MI-MatFact	DMIML	DMIML$_\Sigma$	DMIML$_{\ell_2}$	DMIML$_{\ell_1}$
IAPR-TC	0.711	0.761	0.779	**0.797**	0.788	0.781
Corel 5K	0.691	0.713	0.758	**0.789**	0.773	0.761

Table 2: Average AUC for Corel and IAPR-TC datasets

Corel-5k Last, we evaluate the algorithms on the Corel dataset. The training set was split into training (4000 images) and validation (500 images) and the parameters were tuned on the validation set. After the parameters were tuned, the model was then retrained with that parameter setting on the full dataset (4500 images) and evaluated on the test set.

To ensure fair comparison, we use average AUC for all algorithms even though it is a common practice to use precision and recall for this dataset [4, 9, 12, 17, 19, 20]. We do not do this due to two reasons: 1) The lack of consistency in evaluation protocol of recent advances in image annotation. Given the image annotation literature, there is a wide discrepancy among how the annotations are evaluated. Most works [9, 17, 20] rank the keywords using the learned classifiers, and then assign keywords that achieve top 5 scores to each test image. However a recent work that achieved state-of-the-art results [19] uses a threshold and assigns the keywords if the classifier's score for a given image was above that threshold (thus there may be more than 5 keywords in the annotation which results in higher recall); 2) The lack of consistency in the choice of features. For example [4, 9, 17] use features computed from segments. However, [17] uses features computed from images after partitioning them into rectangles and [20] and [19] use global features. Given these inconsistencies it is not obvious whether the improvement in precision/recall comes from the new features set, or from the number of keywords assigned, or from the learning algorithm itself.

Therefore, we keep the feature set fixed for all the experiments as our goal is to compare the modeling power of the algorithms. We note, however, that we compare our algorithm with the state-of-the-art Matrix Factorization model [19] as trained on the transformed Multiple Instance to single instance problem.

The results for the large-scale dataset show that the best result is achieved using Trace Norm regularization, followed by ℓ_2 regularization then ℓ_1. Both Corel and IAPR-TC have a very large vocabulary and many labels are correlated. Therefore Trace Norm regularization outperforms ℓ_2 regularization, as it takes advantage of label correlation. It is also clear that using Multiple Instance Learning directly is beneficial as even DMIML model with no regularization outperforms MIMLSVM and MI-MatFact. The learning algorithm benefits from using all instances during learning unlike MIMLSVM or MI-MatFact models which lose information during the transformation from Multiple Instance learning to single instance learning. The benefit of Trace Norm regularization is clear not only in case of Multiple Instance Learning but also in case of the transformed Multiple Instance to single instance Learning.

6 Summary and Conclusion

We proposed a solution to Multiple Instance Multiple Label Learning that can be used in the settings where the number of bags and labels is large (such as image annotation). Unlike previous state-of-the-art MIML algorithms our approach does not require approximate probability computation (by using, for example, Gibbs Sampling) or computation of a kernel matrix. Our solution to MIML learning problem is a linear model that is based on a well studied loss-penalty formulation that allows the algorithm to trade-off the loss against the penalty term. The proposed algorithm trains a discriminative model for each possible label in the vocabulary and several regularization terms are considered and compared. The loss function, inspired by the Noisy-Or model for Multiple Instance learning is designed specifically for Multiple Instances. The penalty functions considered are Trace Norm and ℓ_1 and ℓ_2 norms for the classifier weights, and we empirically show strengths and weaknesses of each. We compared the performance of resulting algorithm with several existing approaches to MIML on several image datasets: small but challenging Microsoft visual classes dataset, and two large image datasets. In particular, we considered 2 state-of-the-art algorithms in MIML learning [28, 30] and a state-of-the-art algorithm in Multiple Label learning [2] (and in image annotation [19]) applied to Multiple Instance learning by transforming Multiple Instance learning to single instance learning. We show that our learning algorithm has better performance than the state-of-the art MIML learning algorithms. We also experimentally showed that when the number of labels is small Trace Norm regularization helps find correlations among labels even though overall it performs slightly worse than ℓ_2 regularization. However Trace Norm regularization improves over Frobenius norm on datasets with a large number of labels. In addition, the proposed algorithm, unlike many other state-of-the-art MIML algorithms, is scalable to setting with large number of images and large vocabulary of possible labels.

References

[1] J. Abernethy, F. Bach, T. Evgeniou, and J.-P. Vert. A new approach to collaborative filtering: Operator estimation with spectral regularization. *Journal of Machine Learning Research*, 10:803–826, 2009.

[2] Yonatan Amit, Michael Fink, Nathan Srebro, and Shimon Ullman. Uncovering shared structures in multiclass classification. In *ICML*, 2007.

[3] Andreas Argyriou, Theodoros Evgeniou, and Massimiliano Pontil. Multi-task feature learning. In *NIPS*, 2006.

[4] David M. Blei and Michael I. Jordan. Modeling annotated data. In *SIGIR*, 2003.

[5] Olivier Chapelle. Training a support vector machine in the primal. *Neural Computation*, 19(5):1155–1178, March 2007.

[6] Ronan Collobert, Fabian Sinz, Jason Weston, and Léon Bottou. Trading convexity for scalability. In *ICML*, 2006.

[7] Koby Crammer and Yoram Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research*, 2:265–292, 2001.

[8] Thomas G. Dietterich, Richard H. Lathrop, and Tomás Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence*, 89:31–71, 1997. doi: doi:10.1016/S0004-3702(96)00034-3.

[9] Pinar Duygulu, Kobus Barnard, Nando de Freitas, and David Forsyth. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In *ECCV*, 2002.

[10] H. J. Escalante, C. Hernández, J. Gonzalez, A. López, M. Montes, E. Morales, E. Sucar, L. Villaseñor, and M. Grubinger. The segmented and annotated iapr tc-12 benchmark. *Computer Vision and Image Understanding*, 2009.

[11] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. Liblinear: A library for large linear classification. *Journal of Machine Learning Research*, (9):1871–1874, 2008.

[12] S.L. Feng, R. Manmatha, and V. Lavrenko. Multiple bernoulli relevance models for image and video annotation. In *CVPR*, 2004. doi: 10.1109/CVPR.2004.1315274.

[13] Jorge Sanchez Florent Perronnin and Yan Liu. Large-scale image categorization with explicit data embedding. In *ECCV*, 2010.

[14] Thomas Gärtner, Peter Flach, Adam Kowalczyk, and Alex Smola. Multi-instance kernels. In *ICML*, 2002.

[15] Nadia Ghamrawi and Andrew McCallum. Collective multi-label classification. In *CIKM*, 2005.

[16] J.A. Hanley and B. J. McNeil. The meaning and use of area under a receiver operating characterisitc (roc) curve. *Radiology*, 143:29–36, 1982.

[17] V. Lavrenko, R. Manmatha, and J. Jeon. A model for learning the semantics of pictures. In *NIPS*, 2003.

[18] Dong C. Liu and Jorge Nocedal. On the limited memory method for large scale optimization. *Mathematical Programming*, 45:503–528, 1987.

[19] Nicolas Loeff and Ali Farhadi. Scene discovery by matrix factorization. In *ECCV*, 2008.

[20] Ameesh Makadia, Vladimir Pavlovic, and Sanjiv Kumar. A new baseline in image annotation. In *ECCV*, 2008.

[21] Jorge Nocedal and Stephen Wright. *Numerical Optimization*. Springer, 2000.

[22] Vikas C. Raykar, Balaji Krishnapuram, Jinbo Bi, Murat Dundar, and R. Bharat Rao. Bayesian multiple instance learning: automatic feature selection and inductive transfer. In *ICML*, 2008.

[23] Saharon Rosset, Ji Zhu, and Trevor Hastie. Margin maximizing loss functions. In *NIPS*, 2004.

[24] Shai Shalev-Shwartz, Yoram Singer, and Nathan Srebro. Pegasos: Primal estimated sub-gradient solver for svm. In *ICML*, 2007.

[25] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000. URL `citeseer.ist.psu.edu/shi97normalized.html`.

[26] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *ECCV*, 2006.

[27] Vladimir N. Vapnik. *The nature of statistical learning theory*. Springer-Verlag New York, Inc., New York, NY, USA, 1995. URL http://portal.acm.org/citation.cfm?id=211359.

[28] Sudheendra Vijayanarasimhan and Kristen Grauman. What's it going to cost you?: Predicting effort vs. informativeness for multi-label image annotations. In *CVPR*, 2009.

[29] Paul Viola, John Platt, and Cha Zhang. Multiple instance boosting for object detection. In *NIPS*, 2005.

[30] Zheng-Jun Zha, Xian-Sheng Hua, Tao Mei, Jingdong Wang, Guo-Jun Qi, and Zenfu Wang. Joint multi-label multi-instance learning for image classification. In *CVPR*, 2008.

[31] Min-Ling Zhang and Zhi-Hua Zhou. Multi-instance multi-label learning with application to scene classification. In *NIPS*, 2006. ISBN 0-262-19568-2.

[32] Min-Ling Zhang and Zhi-Hua Zhou. M3MIML: A maximum margin method for multi-instance multi-label learning. In *International Conference on Data Mining*, 2008.