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

Oksana Yakhnenko
http://www.cs.iastate.edu/~oksayakh
Google Inc
New York, NY, USA

Vasant Honavar
http://www.cs.iastate.edu/~honavar
Iowa State University
Ames, IA, USA

Image classification is a challenging task with many applications in computer vision, including image auto-annotation and content-based image retrieval. Recent state-of-the-art image classification and annotation approaches [3, 4] used global features extracted from the images. However, the global features may not be well-suited in when images contain multiple objects, and therefore image classification has been modeled as a Multiple Instance Multiple Label (MIML) learning problem [7, 8, 9]. 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. 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.

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 $f : \mathscr{R}^d \to \mathscr{L}$. The general formulation of learning [6] suggests learning a classifier by trading off between the classifier's average empirical loss and the complexity of the classifier. This formulation has been extended to multiple label learning [2] 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).

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

In case of MIML, the input is a bag of instances $X_i = \left\{x_{i1}, ..., x_{ik_i}\right\}$ and output is a collection of labels $Y_i = \left\{y_i^1, ..., y_i^{m_i}\right\}$. We construct the loss as

$$\text{loss}(y_i^j, h_j(X_i) = -\log(p(y_i^j | X_i))$$

We 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. The loss is then modeled as negative log of the probability of correctly assigning the label:

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

where $\delta(a, b) = 1$ if $a = b$ and 0 otherwise.

The choice of an appropriate penalty function has been an active research area. We consider three loss functions: Trace Norm, Frobenius Norm (defined as $||W||_2^2 = \sum_i w_i^2$) and ℓ_1 Norm (defined as $||W||_1 = \sum_i |w_i|$) .The Trace Norm [1] $\|W\|_\Sigma$ is defined as

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

where $\|\cdot\|_{\mathscr{F}}$ is the Frobenius norm (another name for the matrix ℓ_2 norm). 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 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 [5] including Stochastic Gradient Descent that makes updates for one example at a time.

We use three datasets to evaluate our algorithm and compare it to the state-of-the-art: Microsoft v2, Corel-5K and IAPR TC-12. The results on Microsoft dataset are reported in Table 1.

Method	MIMLSVM	MIMLBoost [8]	MIMIL [8]	MIL-Kernel [7]
Average AUC	0.776 ± 0.02	0.766	0.902	0.896
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

The results on Corel and IAPT-TC datasets are shown in Table 2. We use AUC instead of precision/recall for evaluation of Corel5K since literature that uses this dataset does not use consistent features, or evaluation protocol. Therefore it is not always 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.

	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

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

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

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

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

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

[6] 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`.

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

[8] 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.

[9] 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.