Similarity-based Multi-label Learning

Ryan A. Rossi
Palo Alto Research Center
rrossi@parc.com

Nesreen K. Ahmed
Intel Labs
nesreen.k.ahmed@intel.com

Hoda Eldardiry
Palo Alto Research Center
heldardiry@parc.com

Rong Zhou
Google
rongzhou@google.com

Abstract

Multi-label classification is an important learning problem with many applications. In this work, we propose a principled similarity-based approach for multi-label learning called SML. We also introduce a similarity-based approach for predicting the label set size. The experimental results demonstrate the effectiveness of SML for multi-label classification where it is shown to compare favorably with a wide variety of existing algorithms across a range of evaluation criterion.

1 Introduction

Multi-label classification is an important learning problem [12] with applications in bioinformatics [10], image & video annotation [3, 14] and query suggestions [1]. The goal of multi-label classification is to predict a label vector $\mathbf{y} \in \{0,1\}^K$ for a given unseen data point $\mathbf{x} \in \mathbb{R}^M$.

Previous work has mainly focused on reducing the multi-label problem to a more standard one such as multi-class [9, 2] and binary classification [11], ranking [5] and regression [8, 7]; see [18] for a recent survey. Standard multi-class approaches can be used by mapping a multi-label problem with K labels to a multi-class problem with 2^K labels [9, 2]. Binary classification methods can also be used by copying each feature vector K times and for each copy k an additional dimension is added with value k; and the training label is set to 1 if label k is present and 0 otherwise [11]. Rank-based approaches attempt to rank the relevant labels higher than irreverent ones [5]. Regression methods map the label space onto a vector space where standard regression methods can be applied [8, 7].

In this work, we introduce a similarity-based approach for multi-label learning called SML that gives rise to a new class of methods for multi-label classification. Furthermore, we also present a similarity-based set size prediction algorithm for predicting the number of labels associated with an unknown test instance $\mathbf{x}$. Experiments on a number of data sets demonstrate the effectiveness of SML as it compares favorably to existing methods across a wide range of evaluation criterion. The experimental results indicate the practical significance of SML.

In addition, SML is a direct approach for multi-label learning. This is in contrast to existing methods that are mostly *indirect approaches* that transform the multi-label problem to a binary, multi-class, or regression problem and apply standard algorithms (*e.g.*, decision trees). Furthermore, other rank-based approaches such as RANK-SVM [5] are also indirect extensions of SVM [13, 16] to multi-label classification. Notably, SML completely avoids such mappings (required by SVM) and is based on the more general notion of similarity.

2 Preliminaries

Let $\mathcal{X} = \mathbb{R}^M$ denote the input space and let $\mathcal{Y} = \{1, 2, \ldots, K\}$ denote the set of possible class labels. Given a multi-label training set $\mathcal{D}$ defined as: $\mathcal{D} = \{(\mathbf{x_1}, Y_\mathbf{1}), \ldots, (\mathbf{x}_N, Y_N)\}$ where $\mathbf{x}_i \in \mathcal{X}$ is a M-dimensional training vector representing a single instance and Y_i is the label set associated with $\mathbf{x}_i$. Given $\mathcal{D}$ the goal of the multi-label learning problem is to learn a function $h : \mathcal{X} \to 2^K$ which predicts a set of labels for an unseen instance $\mathbf{x}_j \in \mathbb{R}^M$. A multi-label learning algorithm typically outputs a real-valued function $f : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ where $f_k(\mathbf{x}_i)$ is the confidence of label $k \in \mathcal{Y}$ for the unseen test instance $\mathbf{x}_i$. Given an instance $\mathbf{x}_i$ and its associated label set Y_i, a good multi-label learning algorithm will output larger values for labels in Y_i and smaller values for labels not in Y_i.

We consider a variety of evaluation criterion for comparing multi-label learning algorithms. The multi-label hamming loss is the fraction of incorrectly classified instance-label pairs:

$$\mathbb{E}_\mathcal{D}(f) = \frac{1}{N} \sum_{i=\mathbf{1}}^{N} \frac{1}{K} \Big| \, h(\mathbf{x}_i) \, \Delta \, Y_i \Big| \tag{1}$$

where Δ is the symmetric difference between the predicted label set $\widehat{Y}_i = h(\mathbf{x}_i)$ and the actual ground truth label set Y_i. One-error evaluates how many times the top-ranked label is not in the set of ground truth (held-out) labels:

$$\mathbb{E}_\mathcal{D}(f) = \frac{1}{N} \sum_{i=\mathbf{1}}^{N} \mathbb{I}\Bigg[\, \big[\arg\max_{k \in \mathcal{Y}} f_k(\mathbf{x}_i) \big] \notin Y_i \Bigg] \tag{2}$$

where for any predicate p the indicator function $\mathbb{I}[\, p \,] = 1$ iff p holds and 0 otherwise. Given a set of labels ordered from most likely to least, coverage measures the max position in the ordered list such that all proper labels are recovered:

$$\mathbb{E}_\mathcal{D}(f) = \frac{1}{N} \sum_{i=\mathbf{1}}^{N} \max_{k \in \mathcal{Y}} \; \pi(\mathbf{x}_i, k) - 1 \tag{3}$$

where $\pi(\mathbf{x}_i, k)$ is the rank of label $k \in \mathcal{Y}$. Alternatively, *Ranking loss* measures the fraction of reversely ordered label pairs:

$$\mathbb{E}_\mathcal{D}(f) = \frac{1}{N} \sum_{i=\mathbf{1}}^{N} \frac{1}{|Y_i||\bar{Y}_i|} \Big| \Big\{ (k, k') \in Y_i \times \bar{Y}_i \; \big| \; f_k(\mathbf{x}_i) \leq f_{k'}(\mathbf{x}_i) \Big\} \Big| \tag{4}$$

Average precision measures the average fraction of relevant labels ranked higher than a particular label $k \in Y_i$:

$$\mathbb{E}_\mathcal{D}(f) = \frac{1}{N} \sum_{i=\mathbf{1}}^{N} \frac{1}{|Y_i|} \sum_{k \in Y_i} \frac{\big| \big\{ k' \in Y_i \mid \pi(\mathbf{x}_i, k') \leq \pi(\mathbf{x}_i, k) \big\} \big|}{\pi(\mathbf{x}_i, k)} \tag{5}$$

Multi-label algorithms should have high precision (Eq. 5) with low hamming loss (Eq. 1), one-error (Eq. 2), coverage (Eq. 3), and ranking loss (Eq. 4).

3 Similarity-based Multi-label Learning (SML)

This section presents our general similarity-based approach for multi-label learning called SML. Given a multi-label training set $\mathcal{D} = \{(\mathbf{x_1}, Y_\mathbf{1}), \ldots, (\mathbf{x}_j, Y_j), \ldots, (\mathbf{x}_N, Y_N)\}$ where $\mathbf{x}_j \in \mathbb{R}^M$ is a M-dimensional training vector representing a single instance and Y_j is the label set associated with $\mathbf{x}_j$, the goal of multi-label classification is to predict the label set Y_i of an unseen instance $\mathbf{x}_i \in \mathbb{R}^M$. Assume $w.l.o.g.$ that all feature vectors $\mathbf{x_1}, \ldots, \mathbf{x}_N$ are normalized to length 1. Given the subset $\mathcal{D}_k \subseteq \mathcal{D}$ of training instances with label $k \in \{1, 2, \ldots, K\}$ defined as

$$\mathcal{D}_k = \big\{ (\mathbf{x}_i, Y_i) \in \mathcal{D} \mid k \in Y_i \big\} \tag{6}$$

we estimate the weight $f_k(\mathbf{x}_i)$ of label k for an unseen test instance $\mathbf{x}_i \in \mathbb{R}^M$ as:

$$f_k(\mathbf{x}_i) \;=\; \sum_{\mathbf{x}_j \in \mathcal{D}_k} \Phi \, \langle \mathbf{x}_i, \mathbf{x}_j \rangle \tag{7}$$

where Φ is an arbitrary similarity function. Notably, the proposed family of similarity-based multi-label learning algorithms can leverage any arbitrary similarity function Φ. Furthermore, our approach does not require mappings in high-dimensional Hilbert spaces [15, 6] as required by RANK-SVM [5]. We define a few parameterized similarity functions below. Given M-dimensional vectors $\mathbf{x}_i$ and $\mathbf{x}_j$, the RBF similarity function is:

$$\Phi(\mathbf{x}_i, \mathbf{x}_j) = \exp\Big[-\gamma \, \|\mathbf{x}_i - \mathbf{x}_j\|^{\mathbf{2}}\Big] \tag{8}$$

A common class of similarity measures for vectors of uniform length are polynomial functions:

$$\Phi(\mathbf{x}_i, \mathbf{x}_j) = \Big[\langle \mathbf{x}_i, \mathbf{x}_j \rangle + c\Big]^d \tag{9}$$

where $\langle \cdot, \cdot \rangle$ is the inner product of two vectors, d is the degree of the polynomial, and c is a regularization term trading off higher-order terms for lower-order ones in the polynomial. Linear-SML and quadratic-SML are special cases of Eq. (9) where $d = 1$ and $d = 2$, respectively. Furthermore, all label weights denoted by $f(\mathbf{x}_i)$ for test instance $\mathbf{x}_i$ are estimated as:

$$f(\mathbf{x}_i) \;=\; \begin{bmatrix} f_{\mathbf{1}}(\mathbf{x}_i) \\ \\ \vdots \\ \\ f_K(\mathbf{x}_i) \end{bmatrix} \;=\; \begin{bmatrix} \sum\limits_{\mathbf{x}_j \in \mathcal{D}_1} \Phi \, \langle \mathbf{x}_i, \mathbf{x}_j \rangle \\ \vdots \\ \sum\limits_{\mathbf{x}_j \in \mathcal{D}_K} \Phi \, \langle \mathbf{x}_i, \mathbf{x}_j \rangle \end{bmatrix} \tag{10}$$

After estimating $f(\mathbf{x}_i) = \big[\, f_{\mathbf{1}}(\mathbf{x}_i) \;\; \cdots \;\; f_K(\mathbf{x}_i) \,\big]^T \in \mathbb{R}^K$ via Eq. 10, we predict the label set Y_i of $\mathbf{x}_i$; see Section 3.1 for further details. As an aside, binary and multi-class problems are special cases of the proposed family of similarity-based multi-label learning algorithms. Furthermore, the binary and multi-class algorithms are recovered as special cases of SML when $|Y_i| = 1, \;$ for $1 \leq i \leq N$. Indeed, the proposed similarity-based multi-label learning approach expresses a family of algorithms as many components are interchangeable such as the similarity function Φ, normalization, weighting function Ψ used to control the influence of the individual similarity score S_{ij}, and the sampling or sketching approach to reduce the training data. The expressiveness and flexibility of SML enables it to be easily adapted for application-specific tasks and domains. In addition, SML lends itself to an efficient and straightforward parallel implementation.

3.1 Similarity-based Label Set Prediction

We present a similarity-based approach for predicting the label set size. For each label set Y_i corresponding to a training instance $\mathbf{x}_i$ in the training set $\mathcal{D}$, we set its label to $|Y_i|$, *i.e.*, the number of labels associated with $\mathbf{x}_i$. Let $\mathbf{y} = \big[\, y_{\mathbf{1}} \;\; y_{\mathbf{2}} \;\; \cdots \;\; y_N \,\big] \in \mathbb{R}^N$ denote an N-dimensional label vector where each $y_i = |Y_i|$ is the new transformed cardinality label of $\mathbf{x}_i$ in $\mathcal{D}$. The new label vector $\mathbf{y} \in \mathbb{R}^N$ is used to predict the label set size. In particular, the new training data is: $\mathcal{D}' = \{(\mathbf{x}_i, y_i)\}, \;$ for $i = 1, 2, \ldots, N$ where the label set Y_i of each instance is replaced by its transformed label y_i that encodes the label set size $|Y_i|$ of $\mathbf{x}_i$. Furthermore, let $\mathcal{Y}' = \{|Y_i|\}_{i=\mathbf{1}}^N$ denote the label space given by the transformation and $K' = |\mathcal{Y}'|$ denote the number of unique labels (*i.e.*, label set cardinalities). It is straightforward to see that the above transforms the original multi-label classification problem into a general multi-class problem for predicting the label set size.

Given $\mathcal{D}' = \{(\mathbf{x}_{\mathbf{1}}, y_{\mathbf{1}}), \ldots, (\mathbf{x}_N, y_N)\}$, the label set size of an unseen instance $\mathbf{x}_i$ is predicted as follows. First, the similarity of $\mathbf{x}_i$ with respect to each training instance $(\mathbf{x}_j, y_j) \in \mathcal{D}'$ is derived as $\Phi(\mathbf{x}_i, \mathbf{x}_j), \; 1 \leq j \leq N$ and the similarities from training instances with the same set size (label) $k \in \mathcal{Y}'$ are combined via addition. More formally, the similarity of instances in $\mathcal{D}'$ of the same set size (class label) $k \in \mathcal{Y}'$ with respect to $\mathbf{x}_i$ is:

$$f_k(\mathbf{x}_i) \;=\; \sum_{\mathbf{x}_j \in \mathcal{D}'_k} \Phi \, \langle \mathbf{x}_i, \mathbf{x}_j \rangle \tag{11}$$

where $\mathcal{D}'_k \subseteq \mathcal{D}'$ is the subset of training instances with label $k \in \mathcal{Y}'$. Therefore, we predict the set size of $\mathbf{x}_i$ using the following decision function:

$$\xi(\mathbf{x}_i) = \underset{k \in \mathcal{Y}'}{\arg\max} \sum_{\mathbf{x}_j \in \mathcal{D}'_k} \Phi \, \langle \mathbf{x}_i, \mathbf{x}_j \rangle \tag{12}$$

where $\xi(\cdot)$ is the predicted label set size for $\mathbf{x}_i$. It is straightforward to see that $\xi(\mathbf{x}_i)$ is the label set size with maximum similarity. Given the label set size $\xi(\mathbf{x}_i)$, we predict the label set $\widehat{Y}_i$ of $\mathbf{x}_i$ by ordering the labels from largest to smallest weight based on $f_1(\mathbf{x}_i), f_2(\mathbf{x}_i), \ldots, f_K(\mathbf{x}_i)$ and setting $\widehat{Y}_i$ to the top $\xi(\mathbf{x}_i)$ labels with the largest weight.

Other approaches: Alternatively, we can infer the label set of $\mathbf{x}_i$ by learning a threshold function $t : \mathcal{X} \rightarrow \mathbb{R}$ such that:

$$h(\mathbf{x}) = \Big\{ k \mid f_k(\mathbf{x}) > t(\mathbf{x}),\ k \in \mathcal{Y} \Big\} \tag{13}$$

where $f_k(\mathbf{x})$ is the confidence of label $k \in \mathcal{Y}$ for the unseen test instance $\mathbf{x}$. To learn the threshold function $t(\cdot)$, we assume a linear model $t(\mathbf{x}) = \langle \mathbf{w}, f(\mathbf{x}) \rangle + b$. More formally, we solve the following problem based on the training set $\mathcal{D}$:

$$\underset{\mathbf{w},b}{\text{minimize}} \ \sum_{i=1}^{N} \Big[\langle \mathbf{w}, f(\mathbf{x}_i) \rangle + b - s(\mathbf{x}_i) \Big]^2 \tag{14}$$

In Eq. 14, we set $s(\mathbf{x}_i)$ as:

$$s(\mathbf{x}_i) \ = \ \underset{\tau \in \mathsf{R}}{\arg\min} \ \Big| \{ k \in Y_i \text{ s.t. } f_k(\mathbf{x}_i) \leq \tau \} \Big| \ + \ \Big| \{ q \in \bar{Y}_i \text{ s.t. } f_q(\mathbf{x}_i) \geq \tau \} \Big| \tag{15}$$

where $\bar{Y}_i$ is the complement of Y_i. After learning the threshold function $t(\cdot)$, we use it to predict the label set Y_i for the unseen instance $\mathbf{x}_i$. Nevertheless, any approach that predicts the label set Y_i from the learned weights $f_1(\mathbf{x}_i), \ldots, f_K(\mathbf{x}_i)$ can be used by SML; see [12, 18] for other possibilities.

3.2 Complexity Analysis

Given a single test instance $\mathbf{x}$, the runtime of SML is $\mathcal{O}(NM\bar{K})$ where N is the number of training instances, M is the number of attributes, and $\bar{K} = \frac{1}{N} \sum_{i=1}^{N} |Y_i|$ is the average number of labels per training instance. This is straightforward to see as SML derives the similarity between each training instance's M-dimensional attribute vector. The space complexity of SML for a single test instance $\mathbf{x}$ is $\mathcal{O}(K)$ where K is the number of labels. This obviously is not taking into account the space required by SML and other methods to store the training instances and the associated label sets. For the similarity-based set size prediction approach, the time complexity is only $\mathcal{O}(NM)$ since the label set size with maximum similarity can be maintained in $o(1)$ time. However, the approach uses $\mathcal{O}(K')$ space where $K' \leq K$.

It is straightforward to incorporate a sampling mechanism into the approach to further improve the time and space requirements. In particular, given a new test instance $\mathbf{x}$ we can sample a small fraction of training instances denoted by $\mathcal{D}_s$ via an arbitrary distribution $\mathbb{F}$ and use this smaller set for predicting labels for $\mathbf{x}$.

4 Experiments

This section investigates the practical significance of SML for multi-label classification. In particular, we evaluate SML against a wide variety of multi-label algorithms including:

- ML-KNN [17]: A kNN-based multi-label approach that uses Euclidean distance to find the top-k instances that are closest. ML-KNN was shown to perform well for a variety of multi-label problems.
- BOOSTEXTER [11]: A boosting-based multi-label algorithm called BOOSTEXTER.
- ADTBOOST.MH [4]: A multi-label decision tree approach.
- RANK-SVM [5]: A multi-label SVM approach based on ranking.

For BOOSTEXTER and ADTBOOST.MH we use 500 and 50 boosting rounds respectively since performance did not change with more rounds (which is consistent with [17]). For RANK-SVM we use polynomial kernels with degree 8 which performs best as shown in [5, 17]. Unless otherwise mentioned, our approach uses the RBF similarity function in Eq. (8); the RBF hyperparameter is learned automatically via k-fold cross-validation on 10% of the labeled data. In this work, we systematically compare the multi-label learning algorithms using data from different domains.

4.1 Gene functional classification

The first multi-label learning task we evaluate is based on predicting the functions of genes from Yeast Saccharomyces cerevisiae - a widely studied organism in bioinformatics [10]. Each gene may take on multiple functional classes. In this investigation, we used the Yeast data from [5, 10]. Each gene consists of a concatenation of micro-array expression data and phylogenetic profile data. Following Elisseeff *et al.* [5], we preprocess the data such that only the known structure of the functional classes are used. The resulting multi-label yeast data consists of $N = 2417$ genes where each gene is represented by a 103-dimensional feature vector. There are $K = 14$ labels denoting the functional classes.

Table 1: Experimental results for each multi-label learning algorithm on the yeast data (mean±std).

Evaluation criterion	SML	ML-KNN [17]	BOOSTEXTER [11]	ADTBOOST.MH [4]	RANK-SVM [5]
Hamming loss (↓)	**0.193 ± 0.013**	0.194 ± 0.010	0.220 ± 0.011	0.207 ± 0.010	0.207 ± 0.013
One-error (↓)	**0.220 ± 0.021**	0.230 ± 0.030	0.278 ± 0.034	0.244 ± 0.035	0.243 ± 0.039
Coverage (↓)	**6.082 ± 0.184**	6.275 ± 0.240	6.550 ± 0.243	6.390 ± 0.203	7.090 ± 0.503
Ranking loss (↓)	**0.155 ± 0.011**	0.167 ± 0.016	0.186 ± 0.015	N/A	0.195 ± 0.021
Average precision (↑)	**0.783 ± 0.016**	0.765 ± 0.021	0.737 ± 0.022	0.744 ± 0.025	0.749 ± 0.026

We use 10-fold cross-validation and show the mean and standard deviation. Experimental results for SML and other multi-label learning algorithms are reported in Table 1. Notably, all multi-label algorithms are compared across a wide range of evaluation metrics. The best result for each evaluation criterion is shown in bold. In all cases, our approach outperforms all other multi-label learning algorithms across all 5 evaluation criterion. Furthermore, the variance of SML is also smaller than the variance of other multi-label learning algorithms in most cases. This holds across all multi-label learning algorithms for coverage, average precision, and ranking loss.[1]

4.2 Scene image classification

The second multi-label learning task we evaluate SML for is natural scene classification using image data. In scene classification each image may be assigned multiple labels representing different natural scenes such as an image labeled as a mountain and sunset scene. Therefore, given an unseen image the task is to predict the set of scenes (labels) present in it. The scene data consists of 2000 images where each image contains a set of manually assigned labels. There are $K = 5$ labels, namely, desert, mountains, sea, sunset, and trees. Each image is represented by a 294-dimensional feature vector derived using the approach in [2].

Table 2: Results of the multi-label learning algorithms for natural scene classification (mean±std).

Evaluation criterion	SML	ML-KNN [17]	BOOSTEXTER [11]	ADTBOOST.MH [4]	RANK-SVM [5]
Hamming loss (↓)	**0.140 ± 0.009**	0.169 ± 0.016	0.179 ± 0.015	0.193 ± 0.014	0.253 ± 0.055
One-error (↓)	**0.252 ± 0.026**	0.300 ± 0.046	0.311 ± 0.041	0.375 ± 0.049	0.491 ± 0.135
Coverage (↓)	0.984 ± 0.112	**0.939 ± 0.100**	**0.939 ± 0.092**	1.102 ± 0.111	1.382 ± 0.381
Ranking loss (↓)	**0.164 ± 0.020**	0.168 ± 0.024	0.168 ± 0.020	N/A	0.278 ± 0.096
Average precision (↑)	**0.852 ± 0.016**	0.803 ± 0.027	0.798 ± 0.024	0.755 ± 0.027	0.682 ± 0.093

We use 10-fold cross-validation and show the mean and standard deviation. The experimental results of SML and the other multi-label algorithms using the natural scene classification data are reported in Table 2. The best result for each evaluation criterion is in bold. From Table 2, it is obvious that SML outperforms all other multi-label algorithms on all but one evaluation criterion, namely, coverage. In terms of coverage ML-KNN and BOOSTEXTER are tied and have slightly lower coverage than SML.

[1] Note the ADTBOOST.MH [4] program does not provide ranking loss.

5 Conclusion

We have described a general framework for similarity-based multi-label learning called SML that gives rise to a novel class of methods for the multi-label problem. Furthermore, we also presented a similarity-based approach for predicting the label set size. Experiments on a number of data sets demonstrate the effectiveness of SML as it compares favorably to existing methods across a wide range of evaluation criterion.

References

[1] R. Agrawal, A. Gupta, Y. Prabhu, and M. Varma. Multi-label learning with millions of labels: Recommending advertiser bid phrases for web pages. In *WWW*, pages 13–24, 2013.

[2] M. R. Boutell, J. Luo, X. Shen, and C. M. Brown. Learning multi-label scene classification. *Pattern recognition*, 37(9):1757–1771, 2004.

[3] G. Carneiro, A. B. Chan, P. J. Moreno, and N. Vasconcelos. Supervised learning of semantic classes for image annotation and retrieval. *TPAMI*, 29(3):394–410, 2007.

[4] F. De Comité, R. Gilleron, and M. Tommasi. Learning multi-label alternating decision trees from texts and data. In *MLDM*, volume 2734, page 35. Springer, 2003.

[5] A. Elisseeff and J. Weston. A kernel method for multi-labelled classification. In *NIPS*, pages 681–687, 2002.

[6] C.-W. Hsu and C.-J. Lin. A comparison of methods for multiclass support vector machines. *Transactions on Neural Networks*, 13(2):415–425, 2002.

[7] D. J. Hsu, S. M. Kakade, J. Langford, and T. Zhang. Multi-label prediction via compressed sensing. In *NIPS*, pages 772–780, 2009.

[8] S. Ji, L. Sun, R. Jin, and J. Ye. Multi-label multiple kernel learning. In *NIPS*, pages 777–784, 2009.

[9] A. McCallum. Multi-label text classification with a mixture model trained by em. In *AAAI workshop on Text Learning*, pages 1–7, 1999.

[10] P. Pavlidis and W. N. Grundy. Combining microarray expression data and phylogenetic profiles to learn gene functional categories using support vector machines. In *ICCBB*, 2000. Yeast data.

[11] R. E. Schapire and Y. Singer. Boostexter: A boosting-based system for text categorization. *Machine learning*, 39(2-3):135–168, 2000.

[12] G. Tsoumakas and I. Katakis. Multi-label classification: An overview. *IJDWM*, 3(3), 2006.

[13] V. N. Vladimir. The nature of statistical learning theory, 1995.

[14] C. Wang, S. Yan, L. Zhang, and H.-J. Zhang. Multi-label sparse coding for automatic image annotation. In *CVPR*, pages 1643–1650, 2009.

[15] J. Weston, C. Watkins, et al. Support vector machines for multi-class pattern recognition. *ESANN*, 99:219–224, 1999.

[16] P. Wolfe. A duality theorem for non-linear programming. *Quarterly of applied mathematics*, pages 239–244, 1961.

[17] M.-L. Zhang and Z.-H. Zhou. Ml-knn: A lazy learning approach to multi-label learning. *Pattern recognition*, 40(7):2038–2048, 2007.

[18] M.-L. Zhang and Z.-H. Zhou. A review on multi-label learning algorithms. *TKDE*, 26(8): 1819–1837, 2014.