Multi-Label Learning with Global Density Fusion Mapping Features

Yumeng Guo[1,2], Fulai Chung[2], Guozheng Li[1,3*]
[1]Department of Control Science and Engineering, Tongji University, Shanghai, China
[2]Department of Computing, Hong Kong Polytechnic University, Hong Kong
[3]Data Center of Traditional Chinese Medicine, China Academy of Chinese Medical Science, Beijing, China
{csguoym, cskchung}@comp.polyu.edu.hk, gzli@ndctcm.cn

Abstract

In multi-label learning, each instance is associated with a set of class labels simultaneously. This is a prevalent problem in data analysis. Existing approaches learn from multi-label data by employing original feature space in the discrimination process of all class labels. However, this traditional strategy might be suboptimal as the original feature space exists redundant and irrelevant information, which reduce the performance of classification. In this paper, another strategy to learn from multi-label data is studied, where reconstructed feature space is exploited to boost the classification performance. Accordingly, an intuitive yet effective algorithm named ATOM, i.e. multi-label learning with global density fusion mapping features, is proposed. ATOM firstly reconstructs feature spaces specific to each and no label by conducting clustering analysis on its belonging instances, and then utilizes density fusion to excavate optimum centers from the cluster center union, at last performs classification by querying the reconstructed feature spaces. Comprehensive experiments on a total of 12 benchmark data sets clearly validate the superiority of ATOM against other competitors.

1 Introduction

Multi-label learning is a prevalent problem in many applications of data analysis, where each data instance is assigned with multiple class labels [Tsoumakas *et al.*, 2009]. For example, in image annotation [Cabral *et al.*, 2011], [Cabral *et al.*, 2015], each image may contains multiple classes' objects. In document categorization [Rubin *et al.*, 2012], [Schapire and Singer, 2000], each document may belong to multiple topics. In gene or protein function prediction [Cesa-Bianchi *et al.*, 2012], [Wang and Li, 2013], [Wang *et al.*, 2015], each gene or protein may associated with multiple functions. Multi-label learning aims to build classifiers to handle the complex nature of multi-label objects. Although many multi-label algorithms have well explored the label space structure to improve the classification performance, only focusing on output space, it is not satisfied. To move forward, learning effective multi-label classifiers from feature space is important to be investigated.

During the past decades, many significant multi-label learning approaches have been proposed [Zhang and Zhou, 2014]. One straightforward strategy for multi-label learning is utilizing original feature representation of the instances to discriminated all the class labels by exploring the label space structure (label correlations). Although this strategy has successfully designed many multi-label algorithms [Zhang and Zhou, 2014], it might be straightforward and monotonous. In other words, it might be suboptimal as the original feature space may have redundant or irrelevant information to disturb the classification performance.

In this paper, we propose a novel algorithm named ATOM, i.e. multi-label learning with global density fusion mapping features. Briefly, ATOM learns from multi-label data with three intuitive simplified steps. Firstly, for each and no class label, clustering analysis is performed on its training instances, and then we combine all the cluster centers as a union. Secondly, aimed at efficiently excavating cluster centers reducing redundant and irrelevant information, density fusion technique is employed to update the cluster center union. Thirdly, reconstructed feature spaces based on distance mapping and linear embedding is constructed by querying the final cluster center union. Fourthly, a family of classifiers are induced where each of them is derived from the reconstructed feature space other than the original one.

To well evaluate the performance of the proposed approach, comparative studies over twelve regular-scale and large-scale data sets and six evaluation criteria have been employed in this paper. Experimental results show that: (a) ATOM achieves superior performance against several competitors of multi-label learning algorithms; (b) ATOM's global density fusion mapping features have the potential of being a general strategy to improve multi-label learning algorithms comprising a number of binary classifiers.

The remainder of this paper is organized as follows. We reviews some existing approaches to multi-label learning in Section 2. The proposed multi-label algorithm ATOM is presented in Section 3. We then report the experimental design and results analysis in Section 4. At last, we conclude and discusses several issues for future work in Section 5.

*This research was supported by the Natural Science Foundation of China (61273305).

2 Related Work

Recently, multi-label learning has received rapidly increased attention from machine learning community, due to its widely existing applications in real world. There is a rich body of work on the research of multi-label learning. Generally, we provide a review to the existing approaches, which can be categorized into two classes: problem transformation approaches and algorithm adaptation approaches.

Problem transformation approaches tackle multi-label learning problems into one or more single-label learning problems and ignore the correlations of labels. Thus, many conventional widely existing single-label algorithms can be employed in this area, such as such as support vector machines (SVM) [Boutell *et al.*, 2004], k nearest neighbor (kNN) [Zhang and Zhou, 2007] and decision trees [Clare and King, 2001], etc. For an unseen instance, the final prediction result is derived from the combination of all single-label classifiers' predictions. The major merit of problem transformation approaches lies in their operational flexibility which is combining existing single-label algorithms and conceptual simplicity which can boost the algorithm design. However, due to their ignorance of label correlations, the effectiveness of these approaches might be suboptimal.

Algorithm adaptation approaches tackle multi-label learning directly, which adapt single-label algorithms to multi-label cases. The process of training classifiers and predicting a unseen instance in this kind of algorithms is similar to traditional single-label algorithms. The major merit of algorithm adaptation approaches is that they can utilize the characteristics of a multi-label learning problem in a more concise and elegant way. Specially, these approaches exploiting pairwise (second-order) relationships between labels or high-order relationships among labels. For second-order approaches, they can utilize the ranking criterion, such as support vector machines [Elisseeff and Weston, 2001], neural networks [Loza Mencía and Fürnkranz, 2008], or the co-occurrence patterns, such as [Fürnkranz *et al.*, 2008], [Madjarov *et al.*, 2011]. For high order approaches, they can impose all other class labels influences on each label or part of class labels, label subsets, such as utilizing hypothesis of linear combination [Cheng and Hüllermeier, 2009], nonlinear mapping [Montañés *et al.*, 2014], shared subspace [Ji *et al.*, 2010], randomly selecting the label subsets [Kumar *et al.*, 2012], imposing graph structure to determine the specific label subsets [Zhang and Zhang, 2010], [Guo and Gu, 2011]. Obviously, algorithm adaptation approaches could address strong label correlation to certain extent and thus are more relatively effective than problem transformation approaches, while would be high computational complexities.

A common property of existing approaches is that they handle multi-label learning problem mainly focusing on the perspective of output space, except LIFT [Zhang, 2011], [Zhang and Wu, 2015], where label-specific feature are exploited to benefit the discrimination of different class labels. For most of them, it is unsatisfactory to utilize original feature space to discriminate all the labels. In the next section, we will present the ATOM algorithm which handles multi-label data by reconstructing feature space via global density fusion mapping features.

Algorithm 1 The ATOM Algorithm

Inputs:

$\mathcal{D}$: multi-label training set $\{(\boldsymbol{x}_i, Y_i)|1 \leq i \leq m\}$ $(\boldsymbol{x}_i \in \mathcal{X}, Y_i \subset \mathcal{Y}, \mathcal{X} = \mathbb{R}^d, \mathcal{Y} = \{l_1, l_2, \ldots, l_q\})$

β: ratio parameter as used in Eq. (2)

$\mathfrak{L}$: binary learner for classifier induction

$\boldsymbol{u}$: unseen instance $(\boldsymbol{u} \in \mathcal{X})$

Outputs:

Y: predicted label set for $\boldsymbol{u}$ $(Y \subset \mathcal{Y})$

1: **for** $t = 0$ to q **do**
2: Form $\mathcal{G}_t$ based on $\mathcal{D}$ according to Eq. (1)
3: Perform k-means clustering on $\mathcal{G}_t$, each with m_t clusters as defined in Eq. (2)
4: **end for**
5: Generate final cluster center union with Eq. (3) and Eq. (4)
6: Create the mapping $\phi^{'}$ according to Eq. (5)
7: Create the mapping $\phi^{''}$ according to Eq. (6)
8: Generate the mapping ϕ according to Eq. (8)
9: **for** $k = 1$ to q **do**
10: Form $\mathcal{B}_k$ according to Eq. (9)
11: Induce $\mathfrak{c}_k$ by invoking $\mathfrak{L}$ on $\mathcal{B}_k$, i.e. $\mathfrak{c}_k \leftarrow \mathfrak{L}(\mathcal{B}_k)$
12: **end for**
13: Return Y according to Eq. (10)

3 The ATOM Algorithm

Given a training set $\mathcal{D} = \{(\boldsymbol{x}_i, Y_i)|1 \leq i \leq m\}$ with m multi-label training examples, where $\boldsymbol{x}_i \in \mathcal{X}$ is a d-dimensional feature vector and $Y_i \subseteq \mathcal{Y}$ is the set of relevant labels associated with $\boldsymbol{x}_i$. Then, ATOM learns from $\mathcal{D}$ by taking five elementary detailed steps, i.e. global information extraction, distance mapping features construction, linear representation features construction, fisher's density fusion analysis of reconstructed feature spaces and classification models induction.

3.1 Global Information Extraction

In the first step, ATOM aims to extract global information which could effectively capture the specific characteristics of each and no label, so as to facilitate its discrimination process. Global information means information from inherent properties of the training set with respect to each and no class label. More specifically, for each class label $l_k \in \mathcal{Y}$ and no class label, we divide the training set with reposition into $(q + 1)$ parts: q positive instances sets $\mathcal{G}_k$ $(1 \leq k \leq q)$ and one negative instances set $\mathcal{G}_0$, which correspond to:

$$\begin{aligned} \mathcal{G}_k &= \{\boldsymbol{x}_i|(\boldsymbol{x}_i, Y_i) \in \mathcal{D}, l_k \in Y_i\} \\ \mathcal{G}_0 &= \{\boldsymbol{x}_i|(\boldsymbol{x}_i, Y_i) \in \mathcal{D}, Y_i = \emptyset\} \end{aligned} \tag{1}$$

Intuitively, $\mathcal{G}_t$ $(0 \leq t \leq q)$, defined as globality for each label, consist of training instances with and without label l_k respectively.

To extract global information from $\mathcal{G}_t$, ATOM chooses to employ partitions of $\mathcal{G}_t$, respectively, as the foundation of

reconstructed feature space. Therefore, suppose $\mathcal{G}_t$ is partitioned into m_t disjoint partitions whose centers are denoted as $\mathcal{C}_t = \{\boldsymbol{c}_t^1, \boldsymbol{c}_t^2, \ldots, \boldsymbol{c}_t^{m_t}\}$ ($\mathcal{C}_t \in \mathbb{R}^{d \times m_t}, \boldsymbol{c}_t \in \mathbb{R}^d$). To gain these appropriate partitions, we consider optimizing reconstruction error, respectively, as follows:

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^{m_t} \|\mathcal{C}_t \boldsymbol{s}_i^t - \boldsymbol{x}_i^t\|_2^2 \\ \text{subject to} \quad & \|\boldsymbol{s}_i^t\|_{0,1} = 1, \forall i = 1, \ldots, m_t^g \end{aligned}$$

Here, $\boldsymbol{s}_i^t \in \mathbb{R}^{m_t}$, $\boldsymbol{x}_i^t \in \mathcal{G}_t$, and $m_t^g = |\mathcal{G}_t|$ is the number of positive instances for each and no class label. Here, $|\cdot|$ returns the set cardinality.

However, to gain the centers of partitions, it is hard to be optimized due to the condition $\|\boldsymbol{s}\|_{0,1} = 1$ $(0{-}norm, 1{-}norm)$. As a compromise, the popular k-means clustering algorithm is employed to handle this [Jain *et al.*, 1999]. Although it might be suboptimal due to the centers of initialization and the number of iteration, but it is effective and simple. To mitigate potential risks brought by the class distribution problem, ATOM sets adaptive number of clusters for $\mathcal{G}_t$. In this way, clustering information gained from instances in $\mathcal{G}_t$ are treated with corresponding importance.

Specifically, the number of clusters retained for $\mathcal{G}_t$ is set as follows:

$$m_t = \lceil \beta \cdot m_t^g \rceil \ (0 \leq t \leq q) \tag{2}$$

Here, $\lceil \cdot \rceil$ denotes the retained integer and $\beta \in [0, 1]$ is a ratio parameter controlling the number of clusters being retained.

3.2 Density Fusion for Centers Reduction

In the second step, ATOM aims to implement density fusion for centers reduction. Form the above, we can define the cluster center union $\mathcal{C} = \{\boldsymbol{c}^1, \boldsymbol{c}^2, \ldots, \boldsymbol{c}^{T_o}\}$ ($\mathcal{C} \in \mathbb{R}^{d \times T_o}, \boldsymbol{c} \in \mathbb{R}^d$). Here, the total number of cluster centers T_o is computed as follows:

$$T_o = \sum_{t=0}^{p} m_t$$

Due to the data distribution of the training set and the construction way of G_t, the centers union may exist pairwise centers with no significant difference. Specially, this can affect the performance of utilizing the centers union in the next steps. A good approach which can drop out or fuse some redundant centers is based on the assumptions that significant cluster centers are surrounded by more neighbors and that they are at a relatively large distance from any other cluster centers. For each center, we compute two quantities: its local density p_i with instances and its fused center c_i, which replaces the center i, with any other centers. These quantities depend on the distances $d_{ij_1}^{'}$ $(1 \leq j_1 \leq m)$ between center i and instances and distances $d_{ij_2}^{''}$ $(1 \leq j_2 \leq T_o)$ with density between center i and any other centers, respectively, which are assumed to satisfy the triangular inequality. The local density p_i of center i is defined as follows:

$$p_i = \sum_{j_1=1}^{m} \chi(d_{ij_1}^{'} - d_{c1}) \tag{3}$$

Here, $\chi(x) = 1$ if $x \leq 0$ and $\chi(x) = 0$ otherwise, and d_{c1} is a cutoff distances. Basically, p_i is equal to the number of points that are closer than d_{c1} to center i.

To compute the fused center c_i for center i, first of all, we define the minimize density p_i^{mz} for center i as follows:

$$p_i^{mz} = \min\{p_{j_2}^{mz} \cdot \chi(d_{ij_2}^{''} - d_{c2}) \cdot \chi(p_{j_2} - p_i)\} \tag{4}$$

And then we regard the center $\boldsymbol{c}^{j_2}$ according to p_i^{mz} as the temporary fused center to replace center i. Finally, iteration for previous step until to the convergence, we obtain the final fused center c_i for each center. We gain the updated final cluster center union $\mathcal{C} = \{\boldsymbol{c}^1, \boldsymbol{c}^2, \ldots, \boldsymbol{c}^{T_o^{'}}\}$ ($\mathcal{C} \in \mathbb{R}^{d \times T_o^{'}}, \boldsymbol{c} \in \mathbb{R}^d$).

3.3 Distance Mapping

In the third step, ATOM aims to construct distance mapping features. Intuitively, cluster center union generated by the k-means algorithm and density fusion method characterize the underlying structure of the original feature space, which can be served as appropriate building blocks (prototypes) for the construction of global features. Here, a mapping $\phi^{'} : \mathcal{X} \to \mathcal{Z}^{'}$ from the original d-dimensional input space $\mathcal{X}$ to the $T_o^{'}$-dimensional distance mapping feature space is created as follows:

$$\phi^{'T}(\boldsymbol{x}) = [d^1, d^2, \ldots, d^{T_o^{'}}] \tag{5}$$

Where

$$d^i = \|\boldsymbol{x} - \boldsymbol{c}^i\|_2 \ \ (1 \leq i \leq T_o^{'})$$

Here, we employ the Euclidean distance $(2 - norm)$ as the metric to measure two vectors in this paper.

3.4 Linear Embedding

In the forth step, ATOM aims to construct linear embedding features. Conceptually, the retained cluster centers can also be utilized as the basis of linear reconstructed feature space. Specifically, each instance can be represented as the linear weighted each center in the cluster center union. Here, a mapping $\phi^{''} : \mathcal{X} \to \mathcal{Z}^{''}$ from the original d-dimensional input space $\mathcal{X}$ to the $T_o^{'}$-dimensional linear representation feature space is created as follows:

$$\phi^{''T}(\boldsymbol{x}) = [w^1, w^2, \ldots, w^{T_o^{'}}] \tag{6}$$

Here, w_k^i $(1 \leq i \leq T_o^{'})$ is the reconstructed weight for each center in the cluster center union.

Accordingly, the problem above can be defined as a solution problem as follows:

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^{m} \|\boldsymbol{x}_i - \sum_{j=1}^{T_o^{'}} w_i^j \boldsymbol{c}^j\|_2^2 \\ \text{subject to} \quad & \sum_{j}^{T_o^{'}} w_i^j = 1, \forall j = 1, \ldots, T_o^{'} \end{aligned} \tag{7}$$

Here, $\boldsymbol{c}^j$ is the j-th column of cluster center union $\mathcal{C}$.

Table 1: Characteristics of The Experimental Data Sets

Data set	$\|\mathcal{S}\|$	$dim(\mathcal{S})$	$L(\mathcal{S})$	$F(\mathcal{S})$	$LCard(\mathcal{S})$	$LDen(\mathcal{S})$	$DL(\mathcal{S})$	$PDL(\mathcal{S})$	Domain	URL*
emotions	593	72	6	numeric	1.869	0.311	27	0.046	music	URL 1
genbase	662	1185	27	nominal	1.252	0.046	32	0.048	biology	URL 1
image	2000	294	5	numeric	1.236	0.247	20	0.010	images	URL 3
scene	2407	294	6	numeric	1.074	0.179	15	0.006	images	URL 1
yeast	2417	103	14	numeric	4.237	0.303	198	0.082	biology	URL 3
slashdot	3782	1079	22	nominal	1.181	0.054	156	0.041	text	URL 2
corel5k	5000	499	374	nominal	3.522	0.009	3175	0.635	images	URL 1
rcv1(subset1)	6000	944	101	numeric	2.880	0.029	1028	0.171	text	URL 1
rcv1(subset2)	6000	944	101	numeric	2.634	0.026	954	0.159	text	URL 1
corel16k(sample1)	13766	500	153	nominal	2.859	0.019	4803	0.349	images	URL 1
corel16k(sample2)	13761	500	164	nominal	2.882	0.018	4868	0.354	images	URL 1
mediamill	43907	120	101	numeric	4.376	0.043	6555	0.149	video	URL 1

* URL 1: *http://mulan.sourceforge.net/datasets.html*
URL 2: *http://meka.sourceforge.net/#datasets*
URL 3: *http://cse.seu.edu.cn/people/zhangml/Resources.htm#data*

3.5 Models of Inducing

In the fifth step, ATOM aims to induce a family of q classifiers $\{\mathfrak{c}_1, \mathfrak{c}_2, \ldots, \mathfrak{c}_q\}$ with the generated global density fusion mapping features. From the above, a mapping $\phi : \mathcal{X} \to \mathcal{Z}$ from the original d-dimensional input space $\mathcal{X}$ to the $2T_o^{'}$-dimensional reconstructed feature space is created as follows:

$$\phi^T(\boldsymbol{x}) = [\phi^{'T}(\boldsymbol{x}), \phi^{''T}(\boldsymbol{x})] \tag{8}$$

Here, $\phi(\boldsymbol{x})$ is the global density fusion mapping features for each instances which is the coalition of distance mapping and linear embedding features.

For each class label $l_k \in \mathcal{Y}$, a new binary training set $\mathcal{B}_k$ with m examples is reconstructed from the original multi-label training set $\mathcal{D}$ and the identical mapping ϕ as follows:

$$\mathcal{B}_k = \{(\phi(\boldsymbol{x}_i), Y_i(k) | (\boldsymbol{x}_i, Y_i) \in \mathcal{D})\} \tag{9}$$

Here, $Y_i(k) = +1$ if $l_k \in Y_i$; Otherwise, $Y_i(k) = -1$. Based on $\mathcal{B}_k$ any binary learner $\mathfrak{L}$ can be applied to induce a classifier $\mathfrak{c}_k : \mathcal{Z} \to \mathbb{R}$ for l_k.

Give an unseen instance $\boldsymbol{u} \in \mathcal{X}$, its associated label set is predicted as

$$Y = \{l_k | \mathfrak{c}_k(\phi(\boldsymbol{u})) > 0, 1 \leq k \leq q\} \tag{10}$$

In other words, classification model f_k corresponding to each label l_k can be viewed as the composition of $\mathfrak{c}_k$ and ϕ, i.e. $f_k(u) = [\mathfrak{c}_k \circ \phi](\boldsymbol{u}) = \mathfrak{c}_k(\phi(\boldsymbol{u}))$.

3.6 Illustration

Algorithm 1 illustrates the complete description of ATOM. Given the multi-label training examples, ATOM firstly constructs global density fusion mapping features (steps 1 to 8); After that, a family of q binary classifiers are induced based on the constructed features successively (steps 9 to 12); Finally, the unseen instance is fed to the learned models for prediction (step 13).

In terms of constructing global density fusion mapping features, the process shown in Algorithm 1 (steps 1 to 8) only represents an intuitive high-efficient implementation and does not mean it's the unique possible way to construct them. Actually, the mapping ϕ can be implemented in numerous alternative ways, such as setting different values of β, d_{c1} and d_{c2}, utilizing distance of other types for $d(\cdot, \cdot)$ instead of the Euclidean metric, etc. In terms of classifiers induction, the process shown in Algorithm 1 (steps 9 to 12) is a typical binary relevance approach. The major difference lies that ATOM induces the classifiers with the reconstructed feature space instead of the original feature space.

4 Experiments

4.1 Experimental Data Sets

For the experimental part, we have chosen twelve well-known multi-label data sets. These data sets are from various application domains and provided with multiple characteristics of multi-label. Table 1 summarizes detailed description of all multi-label data sets used in the experiments. Simply ordered by the number of example, six regular-scale data sets (first part, less than 5000) as well as six large-scale data sets (second part, equal to or more then 5000) are included. Furthermore, dimensionality reduction is performed on two text data sets with huge number of features which is more than 47000, including *rcv1(subset 1)* and *rcv1(subset 2)*. Specifically, the top 2% features with highest document frequency are retained. Due to the diversity and characteristics of the employed data sets, experimental result analysis reported in this paper is quite comprehensive which aims at providing a solid basis for assessing the ATOM's effectiveness.

For each data set $S = \{(\boldsymbol{x}_i, Y_i) | 1 \leq i \leq p\}$, we use $|S|$, $dim(S)$, $L(S)$ and $F(S)$ to denote the number of examples, number of features, number of possible class labels, and feature type for S respectively. In addition, several other multi-label properties [Tsoumakas *et al.*, 2009], [Read *et al.*, 2011] are denoted as:

- $LCard(S) = \frac{1}{p}\sum_{i=1}^{p}|Y_i|$: label cardinality which measures the average number of labels per example;
- $LDen(S) = \frac{LCard(S)}{L(S)}$: label density which normalizes $LDen(S)$ by the number of possible labels;

Table 2: Predictive Performance of Each Comparing Algorithm (mean ± std. Deviation) on the Six Regular-Scale Data Sets

Comparing algorithm	*emotions*	*genbase*	*image*	*scene*	*yeast*	*slashdot*
	Average precision↑					
ATOM	**0.8311±0.0298**	0.9983±0.0031	**0.8309±0.0163**	**0.8931±0.0153**	**0.7742±0.0128**	**0.7079±0.0180**
LIFT	0.8237±0.0285	**0.9985±0.0027**	0.8248±0.0164	0.8869±0.0171	0.7693±0.0112	0.6957±0.0146
BR	0.8182±0.0306	0.9983±0.0030	0.7983±0.0169	0.8463±0.0180	0.7596±0.0127	0.6832±0.0162
MLkNN	0.8009±0.0274	0.9910±0.0055	0.7902±0.0131	0.8669±0.0152	0.7632±0.0171	0.5004±0.0166
ECC	0.8213±0.0300	0.9979±0.0041	0.7922±0.0198	0.8564±0.0115	0.7525±0.0122	0.6686±0.0199
	Macro-averaging AUC↑					
ATOM	**0.8639±0.0320**	**0.8694±0.1132**	**0.8654±0.0186**	**0.9503±0.0096**	**0.6998±0.0166**	0.7236±0.0278
LIFT	0.8535±0.0339	0.8684±0.1122	0.8597±0.0195	0.9488±0.0094	0.6913±0.0112	**0.7558±0.0397**
BR	0.8421±0.0296	0.8692±0.1125	0.8316±0.0195	0.9157±0.0110	0.6437±0.0114	0.7433±0.0434
MLkNN	0.8443±0.0261	0.8647±0.1099	0.8309±0.0177	0.9337±0.0087	0.6845±0.0152	0.5306±0.0222
ECC	0.8361±0.0251	0.8656±0.1136	0.8318±0.0181	0.9337±0.0089	0.6700±0.0119	0.7436±0.0436
	Hamming loss↓					
ATOM	**0.1748±0.0159**	0.0015±0.0009	**0.1524±0.0095**	**0.0755±0.0056**	**0.1879±0.0060**	**0.0387±0.0020**
LIFT	0.1849±0.0154	0.0024±0.0015	0.1550±0.0095	0.0782±0.0055	0.1909±0.0060	**0.0387±0.0010**
BR	0.1922±0.0153	**0.0005±0.0004**	0.1768±0.0095	0.1038±0.0078	0.1990±0.0050	0.0399±0.0007
MLkNN	0.1920±0.0241	0.0051±0.0023	0.1706±0.0070	0.0850±0.0073	0.1931±0.0079	0.0519±0.0005
ECC	0.1874±0.0226	**0.0005±0.0004**	0.1783±0.0174	0.0942±0.0064	0.2002±0.0068	0.0413±0.0025
	Coverage↓					
ATOM	**0.2765±0.0306**	0.0130±0.0048	**0.1635±0.0100**	**0.0625±0.0077**	0.4457±0.0102	**0.1031±0.0085**
LIFT	0.2805±0.0467	0.0135±0.0007	0.1684±0.0337	0.0647±0.0108	0.4538±0.0324	0.1048±0.0048
BR	0.2849±0.0475	**0.0129±0.0006**	0.1877±0.0375	0.0888±0.0148	0.4588±0.0328	0.1094±0.0050
MLkNN	0.2965±0.0494	0.0162±0.0008	0.1952±0.0390	0.0785±0.0131	**0.4456±0.0318**	0.1873±0.0085
ECC	0.2789±0.0466	0.0132±0.0007	0.1940±0.0388	0.0816±0.0136	0.4568±0.0326	0.1244±0.0057
	One-error↓					
ATOM	**0.2226±0.0484**	**0.0000±0.0000**	**0.2580±0.0334**	**0.1828±0.0252**	**0.2168±0.0180**	**0.3815±0.0221**
LIFT	0.2310±0.0489	**0.0000±0.0000**	0.2680±0.0323	0.1940±0.0277	0.2226±0.0125	0.4016±0.0159
BR	0.2377±0.0552	0.0015±0.0047	0.3085±0.0293	0.2551±0.0289	0.2226±0.0122	0.4170±0.0212
MLkNN	0.2766±0.0470	0.0121±0.0119	0.3205±0.0215	0.2239±0.0302	0.2400±0.0178	0.6386±0.0202
ECC	0.2478±0.0535	0.0015±0.0047	0.3175±0.0337	0.2426±0.0235	0.2191±0.0102	0.4268±0.0257
	Ranking loss↓					
ATOM	**0.1357±0.0313**	**0.0008±0.0015**	**0.1373±0.0118**	**0.0584±0.0100**	**0.1607±0.0093**	**0.0883±0.0084**
LIFT	0.1412±0.0289	0.0011±0.0022	0.1424±0.0144	0.0611±0.0107	0.1649±0.0093	0.0897±0.0070
BR	0.1453±0.0281	**0.0008±0.0020**	0.1660±0.0157	0.0897±0.0105	0.1715±0.0082	0.0932±0.0067
MLkNN	0.1599±0.0294	0.0028±0.0039	0.1774±0.0162	0.0769±0.0078	0.1654±0.0096	0.1727±0.0097
ECC	0.1415±0.0319	0.0010±0.0022	0.1735±0.0196	0.0807±0.0056	0.1758±0.0080	0.1072±0.0098

Table 3: Predictive Performance of Each Comparing Algorithm (mean ± std. Deviation) on the Six Large-Scale Data Sets

Comparing algorithm	*corel5k*	*rcv1-s1*	*rcv1-s2*	*corel16k-s1*	*corel16k-s2*	*mediamill*
	Average precision↑					
ATOM	**0.2910±0.0065**	**0.6054±0.0044**	**0.6331±0.0032**	0.3049±0.0015	0.2998±0.0041	**0.7044±0.0008**
LIFT	0.2880±0.0048	0.5918±0.0049	0.6180±0.0047	**0.3083±0.0024**	**0.3076±0.0020**	0.7000±0.0021
BR	0.2789±0.0038	0.5511±0.0035	0.5857±0.0024	0.2827±0.0052	0.2766±0.0022	0.5089±0.0020
MLkNN	0.2437±0.0038	0.4502±0.0143	0.4772±0.0083	0.2803±0.0023	0.2727±0.0040	0.6757±0.0018
ECC	0.2528±0.0048	0.5601±0.0052	0.5965±0.0038	0.2925±0.0033	0.2883±0.0026	0.6155±0.0177
	Macro-averaging AUC↑					
ATOM	0.5765±0.0069	0.8977±0.0081	**0.8954±0.0090**	0.6858±0.0029	0.7000±0.0084	**0.7093±0.0201**
LIFT	**0.6058±0.0168**	**0.9018±0.0109**	0.8937±0.0130	**0.6966±0.0048**	**0.7116±0.0033**	0.6395±0.0002
BR	0.5333±0.0180	0.8732±0.0143	0.8803±0.0065	0.6527±0.0034	0.6669±0.0083	0.5085±0.0001
MLkNN	0.4629±0.0069	0.6713±0.0079	0.6779±0.0189	0.5637±0.0027	0.5711±0.0053	0.5097±0.0001
ECC	0.5517±0.0153	0.8607±0.0145	0.8705±0.0097	0.6548±0.0041	0.6648±0.0039	0.5237±0.0002
	Hamming loss↓					
ATOM	**0.0093±0.0000**	**0.0255±0.0001**	**0.0223±0.0002**	**0.0187±0.0000**	**0.0175±0.0000**	0.0313±0.0004
LIFT	0.0095±0.0001	0.0261±0.0002	0.0228±0.0002	0.0188±0.0000	0.0176±0.0000	**0.0308±0.0003**
BR	0.0123±0.0001	0.0266±0.0002	0.0233±0.0002	**0.0187±0.0000**	**0.0175±0.0000**	0.0311±0.0003
MLkNN	0.0096±0.0000	0.0276±0.0005	0.0244±0.0002	0.0188±0.0000	0.0176±0.0000	0.0332±0.0003
ECC	0.0145±0.0001	0.0269±0.0002	0.0240±0.0002	0.0188±0.0000	0.0177±0.0001	0.0383±0.0011
	Coverage↓					
ATOM	**0.2692±0.0124**	**0.1228±0.0012**	**0.1174±0.0029**	**0.3018±0.0020**	**0.2939±0.0031**	**0.1790±0.0038**
LIFT	0.2955±0.0008	0.1285±0.0086	0.1250±0.0022	0.3280±0.0021	0.3169±0.0037	0.1953±0.0017
BR	0.2908±0.0008	0.1473±0.0135	0.1376±0.0035	0.3190±0.0021	0.3106±0.0019	0.5696±0.0037
MLkNN	0.3068±0.0008	0.2342±0.0091	0.2270±0.0044	0.3412±0.0022	0.3342±0.0020	0.1810±0.0018
ECC	0.2969±0.0008	0.1486±0.0153	0.1395±0.0058	0.3264±0.0021	0.3180±0.0019	0.2394±0.0024
	One-error↓					
ATOM	**0.6554±0.0097**	**0.4020±0.0061**	**0.3981±0.0065**	**0.6872±0.0046**	**0.6810±0.0101**	0.1556±0.0050
LIFT	0.6874±0.0192	0.4149±0.0079	0.4107±0.0026	0.6973±0.0076	0.6857±0.0081	**0.1483±0.0036**
BR	0.7700±0.0074	0.4519±0.0080	0.4393±0.0034	0.7229±0.0129	0.7215±0.0068	0.2426±0.0057
MLkNN	0.7442±0.0059	0.5730±0.0184	0.5452±0.0098	0.7384±0.0075	0.7473±0.0047	0.1672±0.0038
ECC	0.7136±0.0092	0.4543±0.0096	0.4269±0.0084	0.7030±0.0113	0.6970±0.0108	0.2023±0.0387
	Ranking loss↓					
ATOM	**0.1148±0.0056**	**0.0488±0.0008**	**0.0486±0.0012**	**0.1534±0.0014**	**0.1488±0.0019**	**0.0528±0.0013**
LIFT	0.1232±0.0039	0.0513±0.0047	0.0518±0.0010	0.1656±0.0037	0.1595±0.0016	0.0576±0.0005
BR	0.1241±0.0047	0.0633±0.0086	0.0617±0.0016	0.1632±0.0020	0.1581±0.0015	0.1499±0.0011
MLkNN	0.1346±0.0047	0.1136±0.0052	0.1139±0.0021	0.1761±0.0018	0.1705±0.0022	0.0544±0.0008
ECC	0.1260±0.0038	0.0606±0.0064	0.0571±0.0021	0.1645±0.0012	0.1587±0.0016	0.0762±0.0033

- $DL(S) = |\{Y|(x, Y) \in S\}|$: distinct label sets which counts the number of distinct label combinations in S;
- $PDL(S) = \frac{DL(S)}{|S|}$: proportion of distinct label sets which normalizes $DL(S)$ by the number of example.

4.2 Evaluation Criteria

To assess the performance of multi-label algorithms from various aspects is essential to consider multiple and contrasting evaluation criteria due to the characteristics of multi-label learning. Thus six popular evaluation criteria are employed, i.e. *average precision, macro-average AUC, hamming loss, coverage, one-error and ranking loss*. For a detailed description of these criteria, refer to [Zhang and Zhou, 2014], [Zhang and Wu, 2015]. In essence, all the six criteria produce their values in the interval $[0, 1]$, with higher values indicating better performance for *average precision and macro-averaging AUC* and worse performance for *hamming loss, coverage, one-error and ranking loss*.

4.3 Multi-label Classifiers

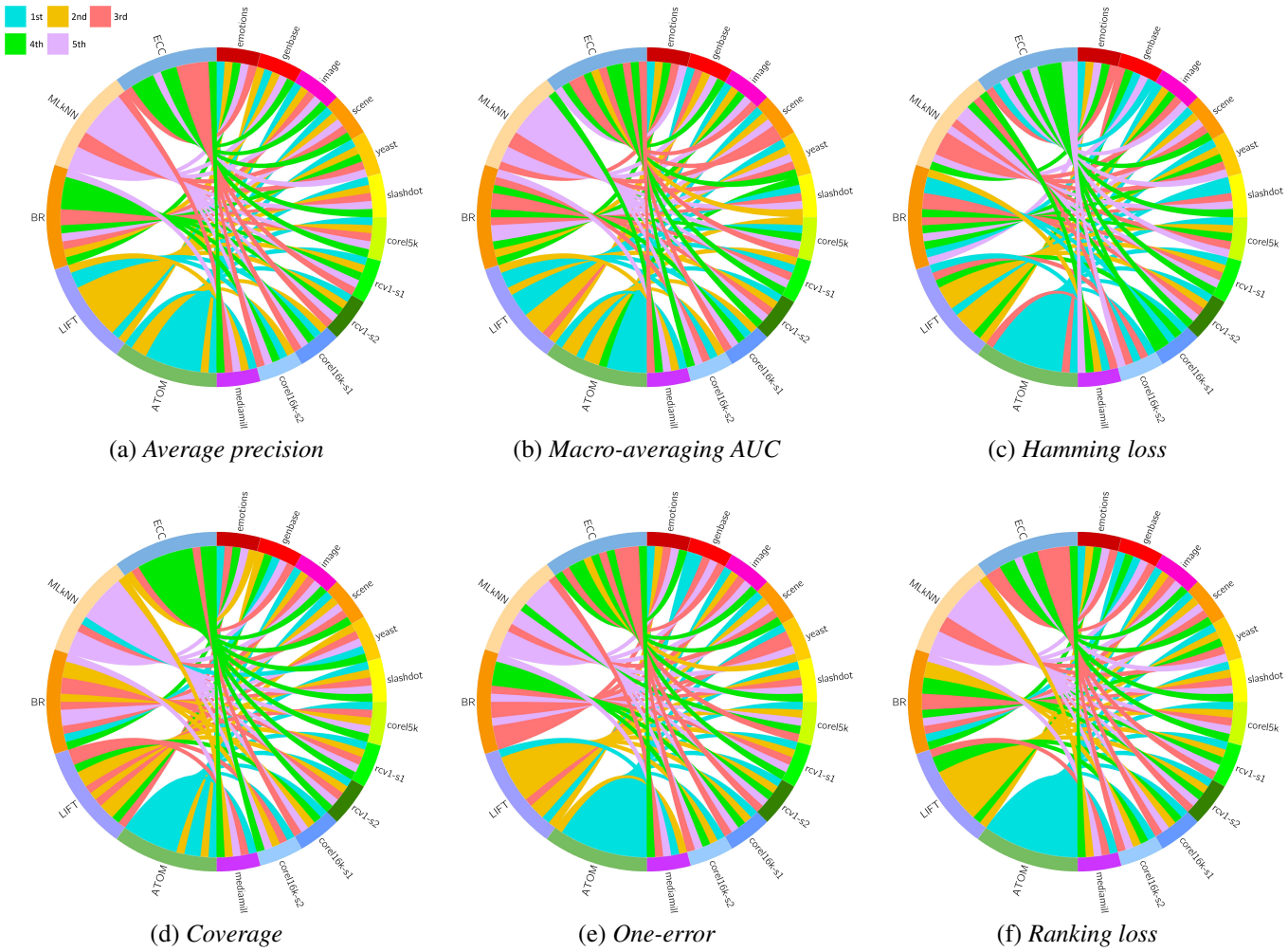
To evaluate the proposed ATOM algorithm, we compare the following four multi-label learning algorithms against ours in the experiments: (1) the label specific features approach, denoted as LIFT [Zhang, 2011], [Zhang and Wu, 2015], which constructs label specific features by utilizing clustering technique on positive and negative instances, and then by querying the clustering results, solves independent binary classification problems for training and testing; (2) the binary relevance approach, denoted as BR [Boutell *et al.*, 2004], which decomposes the multi-label learning problem into independent binary classification problems; (3) the multi-label k nearest neighbors approach, denoted as MLkNN [Zhang and Zhou, 2007], which adapts k nearest neighbors method to handle the multi-label data; (4) the ensemble of classifier chains approach, denoted as ECC [Read *et al.*, 2011], which transforms the multi-label learning problem into a chain of binary classification problems and employs the ensemble learning technique to deal with the classifier chains.

4.4 Experimental Setup

We conduct experiments using ten-fold cross validation on the regular-scale data sets. For the large-scale data sets, we conduct this strategy which is 50% examples randomly sampled as training set and the rest as test set. Then the process is repeated for ten times. For fair comparison, LIBSVM (with linear kernel) [Chang and Lin, 2011] is employed for ATOM, LIFT, BR and ECC as the binary classifier. For ATOM, the ratio β is set to 0.2 and the d_{c1} and d_{c2} depend on the order of magnitudes of number of pairwise distances; For LIFT, the ratio is set to 0.1; For MLkNN, the number of nearest neighbors is set to 10; For ECC, the ensemble size is set to be 10 and the sampling ratio is set to be 50%.

4.5 Results Analysis

The detailed experimental results of each comparing algorithm with six evaluation criteria on the regular-scale and large-scale data sets are demonstrated in Table 2 and Table 3 respectively. For each evaluation criterion, "↑" indicates "the smaller the better" while "↓" indicates "the larger the better". Furthermore, the bold-faced values represent the best performance among all the five comparing algorithms. To visually present the relative performance of ATOM and other comparing algorithms, Fig. 1 illustrates the performance ranking of each



(a) *Average precision* (b) *Macro-averaging AUC* (c) *Hamming loss*

(d) *Coverage* (e) *One-error* (f) *Ranking loss*

Figure 1: Comparison of ATOM against other comparing algorithm under each evaluation criterion. Each data set connects all the algorithms with different color curves simultaneously and the color curves denote the performance ranking of each algorithm corresponding to identical data set.

algorithm corresponding to data sets on each evaluation measure. In each subfigure, each data set connect all the all the algorithms with different color curves simultaneously and the color curves denote the performance ranking of each algorithm corresponding to identical data set.

Across all the 72 configurations (i.e. 12 data sets $\times$ 6 criteria as shown in the two tables and one figure), ATOM ranks in first place among the five comparing algorithms at 81.9% cases. In detail, for the regular-scale data sets, ATOM ranks first in 86.1% cases. And for the large-scale data sets, ATOM ranks first in 77.8% cases. Furthermore, ATOM ranks first in in 79.2% cases on the data sets with spares features (*genbase, slashdot, rcv1-s1 and rcv1-s2*). On the other hand, ATOM ranks first in more than 83.3% cases on the data sets with dense features (*emotions, image, scene, yeast, corel5k, corel16k-s1, corel16k-s2 and mediamill*). These results indicate that ATOM tends to work better in application domains with regular-scale data sets and dense feature representation than those with sparse feature representation.

As shown in Table 2, Table 3 and Fig. 1, ATOM achieves superior performance against BR in terms of each evaluation criterion. Because BR can be regarded as ATOM which keeps the original feature vector untouched, the superior performance of ATOM against BR clearly verifies the effectiveness of employing global density fusion mapping features. ATOM achieves comparable performance against LIFT too. Because LIFT employs the label specific features, this clearly verifies the superior performance of global information. Furthermore, ATOM significantly outperforms MLkNN and ECC. This clearly verifies the effectiveness of reconstructed feature space.

5 Conclusion

The major contribution of our work is to utilize global density fusion mapping features for multi-label learning, which suggests a promising direction for learning from multi-label data. Experiments across the largest number of benchmark data sets up to date show that: (a) ATOM achieves highly competi-

tive performance against other competitors; (b) Multi-label learning algorithms comprising binary classifiers might be improved by utilizing global density fusion mapping features.

In the future, it is interesting to design other global density fusion mapping features generation strategies, incorporate global density fusion mapping features into other multi-label learning algorithms, and improve ATOM by consider label correlations into the global density fusion mapping features construction step.

References

[Boutell *et al.*, 2004] Matthew R. Boutell, Jiebo Luo, Xipeng Shen, and Christopher M. Brown. Learning multi-label scene classification. *Pattern Recognition*, 37(9):1757–1771, 2004.

[Cabral *et al.*, 2011] Ricardo Silveira Cabral, Fernando De la Torre, João Paulo Costeira, and Alexandre Bernardino. Matrix completion for multi-label image classification. In *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011. Proceedings of a meeting held 12-14 December 2011, Granada, Spain.*, pages 190–198, 2011.

[Cabral *et al.*, 2015] Ricardo Silveira Cabral, Fernando De la Torre, João Paulo Costeira, and Alexandre Bernardino. Matrix completion for weakly-supervised multi-label image classification. *IEEE Trans. Pattern Anal. Mach. Intell.*, 37(1):121–135, 2015.

[Cesa-Bianchi *et al.*, 2012] Nicolò Cesa-Bianchi, Matteo Re, and Giorgio Valentini. Synergy of multi-label hierarchical ensembles, data fusion, and cost-sensitive methods for gene functional inference. *Machine Learning*, 88(1-2):209–241, 2012.

[Chang and Lin, 2011] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM TIST*, 2(3):27, 2011.

[Cheng and Hüllermeier, 2009] Weiwei Cheng and Eyke Hüllermeier. Combining instance-based learning and logistic regression for multilabel classification. *Machine Learning*, 76(2-3):211–225, 2009.

[Clare and King, 2001] Amanda Clare and Ross D King. Knowledge discovery in multi-label phenotype data. In *Principles of data mining and knowledge discovery*, pages 42–53. Springer, 2001.

[Elisseeff and Weston, 2001] André Elisseeff and Jason Weston. A kernel method for multi-labelled classification. In *Advances in Neural Information Processing Systems 14 [Neural Information Processing Systems: Natural and Synthetic, NIPS 2001, December 3-8, 2001, Vancouver, British Columbia, Canada]*, pages 681–687, 2001.

[Fürnkranz *et al.*, 2008] Johannes Fürnkranz, Eyke Hüllermeier, Eneldo Loza Mencía, and Klaus Brinker. Multilabel classification via calibrated label ranking. *Machine Learning*, 73(2):133–153, 2008.

[Guo and Gu, 2011] Yuhong Guo and Suicheng Gu. Multi-label classification using conditional dependency networks. In *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011*, pages 1300–1305, 2011.

[Jain *et al.*, 1999] Anil K Jain, M Narasimha Murty, and Patrick J Flynn. Data clustering: a review. *ACM computing surveys (CSUR)*, 31(3):264–323, 1999.

[Ji *et al.*, 2010] Shuiwang Ji, Lei Tang, Shipeng Yu, and Jieping Ye. A shared-subspace learning framework for multi-label classification. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 4(2):8, 2010.

[Kumar *et al.*, 2012] Abhishek Kumar, Shankar Vembu, Aditya Krishna Menon, and Charles Elkan. Learning and inference in probabilistic classifier chains with beam search. In *Machine Learning and Knowledge Discovery in Databases*, pages 665–680. Springer, 2012.

[Loza Mencía and Fürnkranz, 2008] Eneldo Loza Mencía and Johannes Fürnkranz. Pairwise learning of multilabel classifications with perceptrons. In *Proceedings of the International Joint Conference on Neural Networks, IJCNN 2008, part of the IEEE World Congress on Computational Intelligence, WCCI 2008, Hong Kong, China, June 1-6, 2008*, pages 2899–2906, 2008.

[Madjarov *et al.*, 2011] Gjorgji Madjarov, Dejan Gjorgjevikj, and Tomche Delev. Efficient two stage voting architecture for pairwise multi-label classification. In *AI 2010: Advances in Artificial Intelligence*, pages 164–173. Springer, 2011.

[Montañés *et al.*, 2014] Elena Montañés, Robin Senge, José Barranquero, José Ramón Quevedo, Juan José del Coz, and Eyke Hüllermeier. Dependent binary relevance models for multi-label classification. *Pattern Recognition*, 47(3):1494–1508, 2014.

[Read *et al.*, 2011] Jesse Read, Bernhard Pfahringer, Geoff Holmes, and Eibe Frank. Classifier chains for multi-label classification. *Machine Learning*, 85(3):333–359, 2011.

[Rubin *et al.*, 2012] Timothy N. Rubin, America Chambers, Padhraic Smyth, and Mark Steyvers. Statistical topic models for multi-label document classification. *Machine Learning*, 88(1-2):157–208, 2012.

[Schapire and Singer, 2000] Robert E. Schapire and Yoram Singer. Boostexter: A boosting-based system for text categorization. *Machine Learning*, 39(2/3):135–168, 2000.

[Tsoumakas *et al.*, 2009] G Tsoumakas, ML Zhang, and ZH Zhou. Tutorial on learning from multi-label data [http://www. ecml pkdd2009. net/wp-content/uploads/2009/08/learningfrom-multi-label-data. pdf]. In *ECML/PKDD*, 2009.

[Wang and Li, 2013] Xiao Wang and Guo-Zheng Li. Multilabel learning via random label selection for protein subcellular multilocations prediction. *IEEE/ACM Trans. Comput. Biology Bioinform.*, 10(2):436–446, 2013.

[Wang *et al.*, 2015] Xiao Wang, Weiwei Zhang, Qiuwen Zhang, and Guo-Zheng Li. Multip-schlo: multi-label protein subchloroplast localization prediction with chous pseu-

do amino acid composition and a novel multi-label classifier. *Bioinformatics*, page btv212, 2015.

[Zhang and Wu, 2015] Min-Ling Zhang and Lei Wu. Lift: Multi-label learning with label-specific features. *IEEE Trans. Pattern Anal. Mach. Intell.*, 37(1):107–120, 2015.

[Zhang and Zhang, 2010] Min-Ling Zhang and Kun Zhang. Multi-label learning by exploiting label dependency. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, July 25-28, 2010*, pages 999–1008, 2010.

[Zhang and Zhou, 2007] Min-Ling Zhang and Zhi-Hua Zhou. ML-KNN: A lazy learning approach to multi-label learning. *Pattern Recognition*, 40(7):2038–2048, 2007.

[Zhang and Zhou, 2014] Min-Ling Zhang and Zhi-Hua Zhou. A review on multi-label learning algorithms. *IEEE Trans. Knowl. Data Eng.*, 26(8):1819–1837, 2014.

[Zhang, 2011] Min-Ling Zhang. LIFT: multi-label learning with label-specific features. In *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011*, pages 1609–1614, 2011.