

# Active zero-shot learning: a novel approach to extreme multi-labeled classification

Sihong Xie<sup>1</sup>  · Philip S. Yu<sup>2</sup>

Received: 28 October 2016 / Accepted: 9 January 2017 / Published online: 11 February 2017  
© Springer International Publishing Switzerland 2017

**Abstract** Big data bring a huge volume of data in a great speed and in many formats with extremely many labels and concepts to be modeled and predicted, such as in text and image tagging, online advertisement placement, recommendation systems, NLP. This emerging issue of big data is termed “extreme multi-labeled classification” (XMLC) and is challenging due to the time, space and sample complexity in predictive model training and testing. We first define general XMLC and then categorize and review recent methods based on two specific forms of XMLC. We propose a novel method called active zero-shot learning to reduce the above complexities. Since the performance of the unseen class prediction largely depends on the seen classes that have labeled data, we challenge the critical and yet often overlooked assumption that the labeled data can only be passively acquired. We propose a new learning paradigm aiming at accurate predictions of a large number of unseen labels using labeled data from only an intelligently selected small set of seed classes with the help of external knowledge. We further demonstrate that the proposed strategy has desirable probabilistic properties to facilitate unseen classes prediction. Experiments on 4 datasets demonstrate that the proposed algorithm is superior to a wide spectrum of baselines. Based on our findings, we point out several critical and promising future directions in XMLC.

**Keywords** Extreme multi-labeled classification · Active learning · Zero-shot learning

## 1 Introduction

It is not uncommon to have tens of thousands of classes to predict in some real-world classification problems. For example, Amazon products can be classified into some of the many categories; a sizable Twitter dataset readily offers millions of hashtags that can be associated with the tweets; in smart and connected healthcare applications like senior home activity monitoring, many potentially useful human activities (classes) can be captured in data stream from wearable sensors. The common task for these applications is to predict a (usually small) set of relevant labels out of the many labels.<sup>1</sup> This classification problem is termed “extreme multi-labeled classification” (XMLC) and has been studied most recently [3, 19, 28, 38, 40, 41].

XMLC differs from traditional multi-labeled or multi-class classification, in that the number of labels can be on the scale of millions. The distinction makes XMLC more challenging. In the training phase, if one-vs-all multi-labeled classification [30] is adopted, sufficient labeled data have to be collected for each label to train a reasonably good predictive model. However, since there are so many labels, collecting annotated data for all the labels can be extremely time-consuming. Also, in the prediction phase, the time complexity grows linearly with the number of labels and becomes a concern in large-scale online systems like advertisement placement or webpage tagging with millions of labels to be predicted in real time.

---

✉ Sihong Xie  
sxie@cse.lehigh.edu  
Philip S. Yu  
psyu@uic.edu

<sup>1</sup> Department of Computer Science and Engineering, Lehigh University, Bethlehem, PA, USA

<sup>2</sup> Department of Computer Science, University of Illinois at Chicago, Chicago, IL, USA

<sup>1</sup> “Label” and “class” are used interchangeably in this article.

In response to these challenges, two major lines of research have been proposed based on two different assumptions. The first category of methods assumes that certain amount of labeled data are available for each label. Such assumption holds if the data can be labeled by a large number of the users, including user-edited Wikipedia articles, user-tagged tweets, and click-through data. These methods focus on issues such as prediction time complexity and tail labels that consist of most of the labels but have scarce true positives [1, 4–8, 19, 28, 33, 39]. Methods in the second category make the weaker assumption that only a small number of classes have labeled data (the “seen” classes), while the vast remaining classes have zero labeled data (the “unseen” classes) and need to be predicted, resulting in the so-called zero-shot learning problem [12, 14, 15, 22–26, 32, 35]. Both lines of research share the common assumption that the label space is in a low-dimensional space and the original labels can be expressed using a much smaller set of signals, and model training and prediction in this compressed space are thus more efficient.

Regarding the label space compression methods, there are two categories adopted by the XMLC literature, i.e., tree-based (or more generally graph-based) [1, 4, 6, 28, 33, 39] and embedding (or dimension reduction) [2, 5, 7, 8, 13, 18, 20, 21, 36, 38, 41, 42]. Tree-based methods adopt the idea of the traditional decision trees and assume that the labels can be organized in a tree, with each parent node representing a more general class than those represented by its child nodes. The tree thus partitions the label space into disjoint regions represented by leaf nodes. The trees are assumed to be given [4, 6, 33] or learned [1, 3, 11, 17, 28, 39]. Labels are predicted via traversing from the root to a leaf node using parent–child connections, or a node similarity metric using the tree structure (siblings can be taken into account). Since the height of the tree is significantly smaller than the number of original labels, fewer predictions will be needed and in this sense, the label space is compressed. On the other hand, embedding or dimension reduction-based methods project the many labels onto a lower-dimensional space, where each label gets its own coordinates. Depending on different objectives captured, random projection [18], PCA [36], CCA [42], deep neural network [14, 26, 38], hand-crafted features [27], error-correcting codes [12], multi-dimensional scaling [19] and matrix completion [40, 41] have been adopted. We organize the above-mentioned work in Table 1 according to the assumptions and compression approaches these methods adopted.

## 2 XMLC problem definition and methodologies

We first formally define the XMLC problem. Let  $\ell$  be the total number of labels to be predicted, and  $p$  be the num-

**Table 1** Categorization of relevant XMLC literature

Compression methods	Assumptions	
	With labeled data for all labels	Zero-shot
Tree-based	[1, 4, 6, 28, 33, 39]	[23, 25, 27, 32]
Embedding	[2, 5, 7, 8, 13, 18, 20, 21, 36, 38, 41, 42]	[12, 14, 15, 22, 24, 26, 35]

ber of features of the instances. The training data are given by  $\{(\mathbf{x}_1, \mathbf{y}_1), \dots, (\mathbf{x}_n, \mathbf{y}_n)\}$ , where  $\mathbf{x}_i \in \mathcal{X} = \mathbb{R}^p$  and  $\mathbf{y}_i \in \mathcal{Y} = \{0, 1\}^\ell$ . Here, for any  $l \in \{1, \dots, \ell\}$ ,  $y_i^l = 1$  indicates that the  $i$ th instance has the  $l$ th label, and otherwise that the label is irrelevant to the instance. XMLC aims at training classification models for efficient and accurate predictions of all  $\ell$  labels with large  $\ell$  on the test dataset  $\{(\mathbf{x}_{n+1}, \mathbf{y}_{n+1}), \dots, (\mathbf{x}_{n+m}, \mathbf{y}_{n+m})\}$  or in general any sample from  $\mathcal{X} \times \mathcal{Y}$ .

Based on the above definition, two more specific problem settings of XMLC have been studied. In one problem setting (Sect. 2.1), there are labeled data for each label, that is,  $\forall l \in \{1, \dots, \ell\}, \exists i \in \{1, \dots, n\}$ , such that  $y_i^l$  is known. The problem can thus be tackled by traditional multi-labeled classification models, such as binary relevance [7, 8] and one-vs-all [30], which train a classifier for each label. However, a sufficiently large amount of labeled data have to be collected for each label, and the time complexity of training and prediction is linear in the number of labels. Given a large  $\ell$ , even a linear time complexity can be prohibitive. Another problem setting called zero-shot learning (Sect. 2.2) assumes that labeled data for a small number of labels can be used to train models to predict *all* labels. Let the set of  $d$  labels with labeled data be denoted by  $\mathcal{S}$  (the seen classes), and the set of the remaining  $k$  labels be denoted by  $\mathcal{U}$  (the unseen classes). Without loss of generality, assume that the seen classes are indexed by  $\{1, \dots, d\}$ , and the unseen classes are indexed by  $\{d+1, \dots, d+k\}$ . The space of seen labels is  $\mathcal{Y} = \{0, 1\}^d$ , and the space of unseen labels is  $\mathcal{Z} = \{0, 1\}^k$ , with the two spaces being orthogonal. Zero-shot learning predicts the  $k$  unseen classes  $\mathbf{z} \in \mathcal{Z} = \{0, 1\}^k$  using two mappings:  $f: \mathcal{X} \rightarrow \mathcal{Y}'$  and  $g: \mathcal{Y}' \rightarrow \mathcal{Z}$  such that the composed predictive model  $g \circ f: \mathcal{X} \rightarrow \mathcal{Z}$  has good prediction performance on the unseen classes. Here  $\mathcal{Y}'$  can be the same as  $\mathcal{Y}$ , but is a compressed version of  $\mathcal{Y}$  when  $\ell$  and  $k$  are large.

### 2.1 Problem setting 1: XMLC with labeled data for all labels

We review representative methods based on label embeddings and trees in this problem setting. Embedding-based

methods employ various dimension reduction approaches. In [18], compressed sensing is adopted and a random matrix  $A \in \mathbb{R}^{m \times \ell}$  compresses the large label space  $\mathcal{Y}$  to a much lower-dimensional space  $\mathcal{Y}'$ . Linear regression models are learned to map from  $\mathcal{X}$  to  $\mathcal{Y}'$ , where the predictions are then decompressed to  $\mathcal{Y}$  using various algorithms such as orthogonal match pursuit (OMP), CoSaMP and FoBa. In [36], the authors proposed principle label space transformation to improve the above random projection. The left-singular vectors of the label matrix  $\mathbf{Y} \in \{0, 1\}^{\ell \times n}$  are used to compress the labels vectors into  $m$ -dimensional vectors. The decoding is simpler than compressed sensing since the singular vectors are orthonormal. The above two methods only exploit the label matrix  $\mathbf{Y}$  and do not consider the discriminative information available in  $\mathcal{X} \times \mathcal{Y}$ . In [42], the authors proposed to adopt canonical component analysis (CCA) to learn to map both  $\mathcal{X}$  and  $\mathcal{Y}$  to the same lower-dimensional space, such that for each training data  $(\mathbf{x}, \mathbf{y})$ , the image of  $\mathbf{x}$  is highly correlated with that of  $\mathbf{y}$  under the mapping. Besides, the authors argued for the systematic codes [9] that include both the original labels and their images under the CCA mapping. Although such coding scheme is infeasible for scenarios with a large number of labels, the CCA-based encoding merits an independent direction in XMLC research [38]. Bloom filter [8] is a coding scheme that first partitions all labels into  $P$  clusters such that any label vector in the training data can only be contained in at most one cluster. Then, labels in each cluster are coded using a  $K$ -sparse vector of length  $Q$ , such that  $P < \binom{Q}{K}$ . Notice that bloom filter uses only information of the labels.

Tree-based methods partition the label space into disjoint smaller regions. The complexities of coding and encoding can be reduced from  $O(\ell)$  to the scale of  $O(\log(\ell))$  given a balanced tree. Based on different splitting criteria, there are various encoding (tree construction) schemas. In [28], the authors use an ensemble of trees for encoding and decoding. Specifically, a tree is built by splitting the training instances in a top-down manner. The split of a node is determined by minimizing the ranking losses of the rankings in the two resulting children nodes, with splitting uncertainty taken into account. In [39], the authors proposed to partition the input space  $\mathcal{X}$  into subregions, each of which is assigned a small number of relevant labels. During testing time, an instance is first mapped to a single region, and the predictive models for those labels in that region give the relevance scores of the labels. In this way, the cost of predicting all labels can be avoided. In [1], the authors proposed a multi-labeled random forest approach similar to that in [39] to handle millions of labels. Besides these embedding and tree-based approaches, there is a special case where the two kinds of approaches meet. In [3], the authors further reduce prediction time complexity in tree-based approaches, by traversing the learned

tree in an embedded space, which has much lower dimensionality ( $< \log(\ell)$ ) than the original  $d$ .

## 2.2 Problem setting 2: zero-shot learning

The above embedding and tree-based methods assume that labels can be collected for each label for model training. This assumption becomes less prohibitive when there are many, possibly millions, labels. For example, in the health-care application where one wants to identify many potentially useful human activities (classes) in videos, it is markedly laborious to tag the videos for all possible activities. Another example is online advertisement bidding [1], where it is impossible for human annotators to go through millions of labels and training examples (webpages) to identify positive and negative instances for each label. Instead, they resorted to noisy and biased labels automatically inferred from the logs of a search engine. Indeed, their experiments showed that by preprocessing the harvested labels, the performance can be further improved.

Zero-shot learning approaches make an assumption on the other extreme: No labeled data can be collected for the majority of the labels (the “unseen” labels), while a certain amount of labeled data are available for a small number of labels (the “seen” labels). Additionally, external knowledge bases describing class relations, such as a large corpus (e.g., the Internet), domain knowledge (e.g., known attributes of the unseen classes) or ontology (e.g., WordNet), are assumed available to turn the predicted seen labels into predictions of the unseen classes [22, 24, 26]. For example, one can first collect video segments for a small subset of primitive human activities, such as simple movements of body parts, to learn a video-to-movement mapping. Then, more complex composite activities can be identified using the predicted primitive movements via domain knowledge (a composite activity consists of multiple primitive movements).

Zero-shot learning approaches can similarly be categorized into two groups based on the knowledge bases (an embedding space or a tree of labels) adopted. For embedding-based approaches, direct attribute prediction (DAP) and indirect attribute prediction (IAP) [22] are the two fundamental paradigms. More sophisticated zero-shot models are also proposed, such as max-margin semi-supervised learning for exploiting the unlabeled data [23], and multi-view zero-shot learning for utilizing multiple data sources [15]. Multiple knowledge bases such as Wikipedia [14, 26], web search logs [25] and human-annotated images [22] are compared. The authors in [14, 26, 35] propose to learn the intermediate attributes using deep learning. For tree-based approaches, the authors in [32] proposed three similarity metrics on trees to predict unseen classes from seen class predictions. WordNet is such a tree of labels, with each word being a class, and

the classes are connected via hyponym–hypernym relationships. The prediction of an unseen class on a leaf node can be obtained by averaging the predictions of the hypernyms of that unseen class, or the cost-sensitive averaging of the predictions from all leaf nodes of seen classes [10].

### 3 Further reduce the labeling cost via active zero-shot learning

Although the zero-shot learning literature has addressed some of the crucial issues, it assumes that the zero-shot models can only passively learn from labeled data collected for a pre-defined subset of seen labels [23,26]. That is, labeled data are available for the given seen classes but not for the unseens, and a zero-shot learning algorithm has to predict unseen classes using the given labeled data and dependencies among labels. Due to the complex dependencies between seen classes and unseen classes, different seen classes provide varied predictive information for the unseen classes. When a good selection of seen classes is not provided, or does not provide sufficient information (e.g., too few seen classes), we need to decide for which classes labeled data should be collected to predict unseen classes well. In other words, the splitting of all classes into seen and unseen sets of classes is a parameter to optimize in zero-shot learning, while none of the previous zero-shot learning methods has addressed the problem.

We contribute to this class splitting problem and propose to actively and intelligently select a parsimonious set of core classes to collect labeled data, and keep the large number of remaining classes unseen to save labeling efforts. Traditional multi-labeled active learning algorithms are less relevant here, as they assume that for *each* and every label, certain labeled data have to be queried [29,37]. We propose to select the labels as seen ones that can provide most information regarding the unseen ones and characterize such informativeness of a candidate class via the entropy of inter-class similarities. We empirically show that the inter-class similarity follows a beta distribution, based on which we reveal the relationship between the entropy and the probability that an unseen class is sufficiently connected to the seen ones, thus justify the proposed class selection criterion.

#### 3.1 Problem formulation

Since we focus on the effects brought by a class split, we fix the following components of zero-shot learning. We adopt the DAP [22] paradigm and want to select  $d$  classes as seen classes to form the compressed space  $\mathcal{Y}'$ , such that  $d$  is small to minimize labeling efforts, and the prediction for the unseen  $k$  classes is optimized. Logistic regression is adopted to learn the mapping  $f$  from  $\mathcal{X}$  to  $\mathcal{Y}'$ . A class similarity matrix  $K$  is

derived from a related corpus as the knowledge base (see Sect. 5). One can view  $K$  as the adjacent matrix of the graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ , where  $\mathcal{V}$  is the set of all classes and the edge weights are the class similarities. Given two index sets  $\mathcal{I}$  and  $\mathcal{J}$ , let  $K^{\mathcal{I}\mathcal{J}}$  be the sub-matrix of  $K$  that consists of the rows indexed by  $\mathcal{I}$  and columns indexed by  $\mathcal{J}$ . Then  $K^{\mathcal{U}\mathcal{U}}$  is the similarity matrix for the unseen classes, and  $K_{ij}^{\mathcal{U}\mathcal{S}}$  is the similarity between the  $i$ th unseen class and the  $j$ th seen class. With  $\hat{\mathbf{y}} \in \mathbb{R}^d$  being the predicted seen classes for  $\mathbf{x}$ , the mapping from  $\mathcal{Y}'$  to  $\mathcal{Z}$  is  $g: \hat{\mathbf{y}} \mapsto K^{\mathcal{U}\mathcal{S}}\hat{\mathbf{y}}$ .

#### 3.2 Methodology

We propose to iteratively add from the pool of unseen classes more labels that are informative about the remaining unseen classes. The connectivities between the classes can be indicators of information about one class carried by others. Specifically, the connectivity between the  $i$ th unseen class and the other unseen classes can be measured by various centrality metrics of the corresponding  $i$ th node on the sub-graph of  $\mathcal{G}$  consisting of all unseen classes. For example, the degree centrality of the  $i$ th unseen class can be calculated as  $\sum_{j=1}^k K_{ij}^{\mathcal{U}\mathcal{U}}$ , where  $k$  is the current number of unseen classes. We call this strategy “*max-deg-uu*” as it selects the unseen class with the maximal degree. This selection strategy does not consider the distribution of the class similarities between class  $i$  and others: class  $i$  can be strongly connected to only a few unseen classes with high weights, but barely so to the remaining majority classes. Such a class can still have a high degree, but does not add much information about the remaining unseen classes.

Instead, we use entropy to characterize how the connectivities  $K_{ij}^{\mathcal{U}\mathcal{U}}$ ,  $j = 1, \dots, k$  distribute. First, the similarities in  $K^{\mathcal{U}\mathcal{U}}$  are normalized to a probability distribution:

$$\bar{K}^{\mathcal{U}\mathcal{U}} = \text{diag}(\mathbb{1}^\top K^{\mathcal{U}\mathcal{U}})^{-1} K^{\mathcal{U}\mathcal{U}}, \quad (1)$$

where  $\text{diag}(\mathbf{v})$  denotes the diagonal matrix with diagonal elements being the entries of the vector  $\mathbf{v}$ , and  $\mathbb{1}$  is the all-one vector. Then, we calculate the entropy of  $K_{ij}^{\mathcal{U}\mathcal{U}}$ ,  $j = 1, \dots, k$  for the  $i$ th unseen class:

$$H(i) = - \sum_{j=1}^k \bar{K}_{ij}^{\mathcal{U}\mathcal{U}} \log \bar{K}_{ij}^{\mathcal{U}\mathcal{U}}, \quad i = 1, \dots, k. \quad (2)$$

We select the top  $c$  classes that have the highest entropies and move them from  $\mathcal{U}$  to  $\mathcal{S}$ . The labels of the training instances for the selected classes are queried and used to train a model for each of those  $c$  classes. These models are then added to  $f$ . This cycle of selection, querying and training are repeated until the labeling budget runs out. Lastly, we obtain the zero-shot model  $g \circ f$ . The algorithm is shown in Algorithm 1.

**Algorithm 1** Active zero-shot learning

**Input:** Unlabeled training data  $\{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ , class similarity matrix  $K$ , budget  $b$ .  
**Output:** Zero-shot prediction model  $g \circ f$ .  
 Randomly select seeding classes to form  $\mathcal{S}$ , and let  $\mathcal{U} = \{1, \dots, d + k\} \setminus \mathcal{S}$ .  
**while**  $b > 0$  **do**  
     Normalize the sub-matrix  $K^{\mathcal{US}}$  by Eq. (1).  
     Calculate the entropies using Eq. (2).  
     Select  $c$  classes, denoted by  $\mathcal{T}$ , that have the largest entropies.  
     Query the labels of  $\mathcal{T}$  for each training instance.  
      $\mathcal{S} = \mathcal{S} \cup \mathcal{T}, \mathcal{U} = \mathcal{U} \setminus \mathcal{T}, b = b - c \times n$ .  
**end while**  
 Train classification model  $f$  for the labels in  $\mathcal{S}$ .  
 Return  $g \circ f$  with  $g = K^{\mathcal{US}}$ .

**4 Theoretical justification**

The seen–unseen class split has effects on the resulting mappings  $f$  and  $g$ . We study the effects on  $f$  in the experimental sections, and here we compare the effects that the proposed strategy and *max-deg-uu* have on the linear mapping  $g : \mathbf{y} \mapsto K^{\mathcal{US}}\mathbf{y}$ . The prediction of the  $i$ th unseen class is given by  $K_{i:}^{\mathcal{US}}\mathbf{y}$ . We view the entries of  $K^{\mathcal{US}}$  as random variables and analyze the patterns in which the significant values in  $K^{\mathcal{US}}$  distribute. Suppose the  $i$ th unseen class is only associated with the seen ones through insignificant coefficients, then the unseen class predictions through the linear model  $K_{i:}^{\mathcal{US}}$  are less confident. Also, if the unseen class is only related to a few seen classes, even if the connections are strong, the resulting prediction can be misled due to the limited number of seen classes, whereas the unseen class can actually be related to more classes that are not selected into  $\mathcal{S}$ . We would like to select seen classes such that they can sufficiently convey information for most of the unseen classes.

**Definition 4.1** ( *$\delta$ -Covered unseen class*) An unseen class, say the  $i$ th one, is  $\delta$ -covered by the selected seen classes if at least one entry in the row  $K_{i:}^{\mathcal{US}}$  has magnitude at least  $\delta$ .

If an unseen class is not  $\delta$ -covered, then all the seen classes do not carry significant information about the unseen class. Let  $C_\delta(\mathcal{S})$  be the set of unseen classes  $\delta$ -covered by  $\mathcal{S}$ ,  $C_\delta(\mathcal{S}) = \cup_{t_j \in \mathcal{S}} C_\delta(t_j)$ , where  $C_\delta(t_j)$  is the set of unseen classes  $\delta$ -covered by the seen class  $t_j \in \mathcal{S}, j = 1, \dots, d$ . Let  $t_i$  be the  $i$ th unseen class:

$$\begin{aligned} \Pr\{t_i \notin C_\delta(\mathcal{S})\} &= 1 - \Pr\{t_i \in C_\delta(\mathcal{S})\} \\ &= 1 - \Pr\{t_i \in \cup_{j \in [s]} C_\delta(t_j)\} \geq 1 - \sum_{j \in [s]} \Pr\{t \in C_\delta(t_j)\} \\ &= 1 - \sum_{j \in [s]} \Pr\{\bar{K}_{ij}^{\mathcal{US}} \geq \delta\}, \end{aligned}$$

where the last inequality follows from the union bound and the last equality from the definition of  $\delta$ -coverage. The above

provides a lower bound of the probability that an unseen class is not  $\delta$ -covered by any selected seen class. The more and larger the quantities  $\Pr\{t \in C_\delta(t_j)\}$  (or equivalently  $\Pr\{\bar{K}_{ij}^{\mathcal{US}} \geq \delta\}$ ), the smaller the lower bound. For different seen–unseen class splits, the distribution of  $\bar{K}_{ij}^{\mathcal{US}}$ , and thus  $\Pr\{\bar{K}_{ij}^{\mathcal{US}} \geq \delta\}$  will be different. Given  $\delta > 0$  (usually a small value), we want to pick  $\mathcal{S}$ , such that  $\bar{K}_{ij}^{\mathcal{US}}$  are samples from a probability distribution that makes  $\Pr\{\bar{K}_{ij}^{\mathcal{US}} \geq \delta\}$  large, or equivalently, whose cumulative distribution function (CDF) of  $K_{ij}^{\mathcal{US}}$  has its major mass at the upper end.

Regarding the distribution of  $\bar{K}_{ij}^{\mathcal{US}}$ , we find out that these coefficients can be fitted quite well by the beta distribution  $BETA(\alpha, \beta)$ , where  $\alpha$  and  $\beta$  are shape parameters, see Fig. 1a for an example on the unix dataset. How does the entropy guide us to a more desirable beta distribution of the coefficients? We collect the empirical entropies defined by Eq. (2) for each seen class and estimate the shape parameters of the distribution defined by Eq. (1). We find out that the entropies are correlated with the fitted shape parameter  $\alpha$ : entropy grows as  $\alpha$  goes up, see Fig. 1b. Note that the entropy is less correlated to  $\beta$  (Fig. 1c). Similar observations are obtained on the other datasets. Therefore, we fix  $\beta$  and plot two beta distributions with  $\alpha = 0.1$  and  $\alpha = 50$  in Fig. 1d. We can see from the figure that, with a larger  $\alpha$  (the red solid line), the beta distribution has more mass at the upper end, and thus, more samples  $\bar{K}_{ij}^{\mathcal{US}}$  from that distribution will be significant values. As a result, the lower bound of the chance that an unseen class is *not*  $\delta$ -covered is small. On the other hand, with a smaller entropy and  $\alpha$ , there can be a significant mass at the lower end (the blue dotted line).

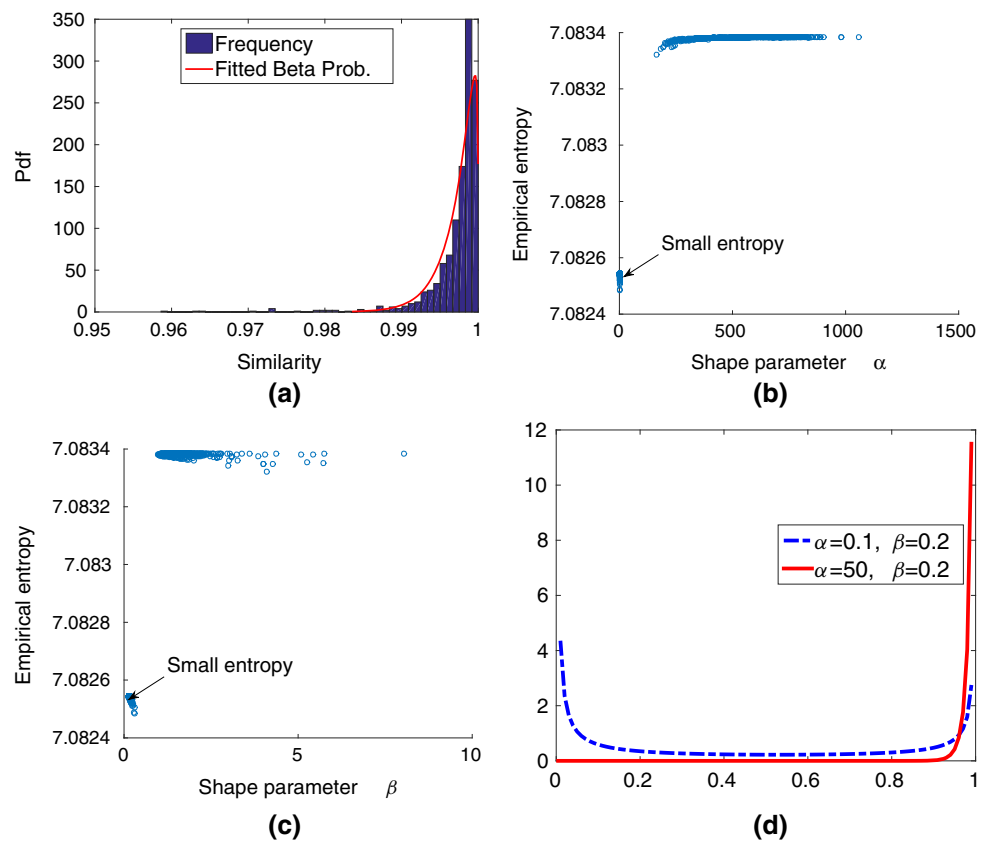
The classes that are uniformly similar to many unseen classes but with low seen–unseen similarity can have high entropies after row normalization Eq. (1), *max-ent-uu* may be misled to pick such classes. Such seen classes will provide little discriminative information, as it is harder to tell from this class that which unseen classes are more relevant. We show in the experiments that the selected seen classes actually have desirable similarities distributions over unseen ones and carry discriminative information.

**5 Experiments**

**5.1 Experimental settings**

StackExchange consists of multiple questions and answers (QA) sub-systems, where the members can ask and answer questions. Users may provide tags to their questions, and by prompting the users to associate their questions with relevant tags, the QA system can increase the tag quality and completeness and facilitate question organization and retrieval. We adopt four sub-systems from StackExchange: askubuntu,

**Fig. 1** Similarity modeled as a beta distribution. Empirical entropy is positively correlated to the shape parameter  $\alpha$  of the distribution. **a** Fitted beta distribution on the unix dataset. **b** Correlation between  $\alpha$  and entropy. **c** Correlation between  $\beta$  and empirical entropy. **d** Two beta distributions with different  $\alpha$  and the same  $\beta$



dba, superuser and unix. The statistics of the four datasets are given in Table 2. Bag-of-words representation with TF-IDF transformation is used to obtain the feature vectors of the questions. Each tag is treated as a class, and a question can have multiple tags, so the tasks can be formulated as multi-labeled classification problems. Only those tags that appear in at least 10 questions are kept. Tags provided by the users for the questions are used as ground truth. Each selection strategy is tested on 20 randomly picked seeding seen classes, and we report the averaged performances over 20 runs. Questions are randomly split into disjoint training and test sets. The training data is used to train classification models (Liblinear with default settings), each of which maps from features to a seen tag. Then, we map the predicted seen classes on the test data to unseen classes via a similarity graph of the tags. In our experiments, we embed all tags in a low-dimensional space via restricted Boltzmann machine trained on the text corpus of questions [34]. Tag similarity is calculated through the kernel function:  $K(\mathbf{t}_1, \mathbf{t}_2) = \exp(\|\mathbf{t}_1 - \mathbf{t}_2\|^2 / \sigma^2)$  where  $\mathbf{t}_1$  and  $\mathbf{t}_2$  are the low-dimensional representations of two tags, and  $\sigma = 10$  throughout the experiments.

We adopt Precision@5 and NDCG@5 as metrics to verify the relevance of the top 5 retrieved tags. Since there is no previous study on active zero-shot learning, we compare *max-ent-uu* with the following baselines that capture different aspects of seen–unseen class splits.

**Table 2** Datasets

	Askubuntu	Dba	Superuser	Unix
# Training	55,684	12,070	93,106	23,069
# Test	55,883	12,211	93,182	23,025
# Tags	1003	345	1895	775

- *max-deg-uu*: as mentioned in Sect. 3.2, this method labels data for the classes that have the highest degree centrality.
- *min-deg-us*: We take the row sums of the matrix  $K^{US}$ , which captures the total similarity between the current unseen tags and seen tags. The unseen class with smallest row sum is picked. The rationale is that unseen tags that are farthest away from the current seen classes can provide complementary information.
- *uncertainty*: This method queries for the training data the top unseen classes that have the highest entropies in their predictions on the training data, according to the current zero-shot prediction model. This baseline runs in an incremental manner as *max-ent-uu* and *min-deg-us*.
- *matrix*: in [16] the author proposed a matrix partition algorithm to split a set of instances into two, such that the mutual information between the distributions of the two sets is maximized. This method is considered to be a representativeness-based active learning method. We

adapt their model and treat classes as instances. This algorithm runs in batch mode and we only report its performance when 100 additional classes are selected.

We set the number of unseen classes selected in each iteration to 2 ( $c = 2$ ) in Algorithm 1 and the other iterative baselines. We test other values for this parameter ( $c = 5$  and  $c = 10$ ) and find out that  $c = 2$  gives the best results.

## 5.2 Results

In Fig. 2, along with the performance of the batch-mode method *matrix*, we show how the zero-shot prediction performances of three iterative algorithms evolve as more labeled data are added. Each row in Fig. 2 consists of two sub-figures showing the performance in Precision@5 and NDCG@5, respectively. In each sub-figure, the performance of *max-ent-uu* (shown in green solid lines) is compared with those of the four baselines. From the figures, we can see that across all datasets and all metrics, *max-ent-uu* consistently outperforms all the baselines. In some cases, *max-ent-uu* ends up with performance two times better than the runner-up (see Fig. 2b, g). Interestingly, *min-deg-us*, *uncertainty*, and *matrix* consistently have medium performance compared with *max-ent-uu* and *max-deg-uu* in all datasets using both metrics when the iterations finish.

Surprisingly, the seemingly naive method *min-deg-us* can gradually pick up its performance and ends up with similar or better performance with the more sophisticated methods *matrix* in the dba and unix datasets. Our explanation is that by selecting the classes that are least similar to the already picked ones, more information can be revealed. However, this baseline fails to consider unseen class coverage information, and the selected classes may not be well connected to the large clusters of unseen classes (as we will see next), leading to less effective seen-to-unseen class mapping. Furthermore, the performance of *uncertainty* is quite close to *matrix* in all cases. Our conjecture is that by picking the current unseen classes that do not have confident predictions, *uncertainty* is able to explore the class space that has not been explored before and ends up with a seen class space that represents the whole class space quite well, which is what *matrix* aims for.

## 5.3 Empirical analysis of *max-ent-uu*

In active zero-shot setting, before one queries the labels of the data for a class, it is difficult to gauge the prior and posterior probability distributions of the class. The only information available is the class similarity from external knowledge bases. Below we empirically show, in two aspects, that even with such a lack of information, *max-ent-uu* is able to pick the informative unseen classes for label queries.

In Fig. 3a, we plot the CDFs of the frequencies of the selected classes that appear in the training instances (namely document frequencies) on the superuser dataset (best viewed in color). We can see that among the five strategies, the *max-deg-uu* tends to select classes that have higher document frequencies than those selected by *max-ent-uu*, as the CDF of *max-deg-uu* is more shifted to the right. It has been shown in text classification that, the more frequent a word appears in the corpus, the less informative it is, as evidenced by the commonly used TF-IDF transformation [31]. A frequent seen class is likely to be predicted more often by predictive models that take the class prior distribution into account. Such a seen class becomes a less discriminative feature when used as features in the mapping  $g$ . This partly explains why the baseline *max-deg-uu* has the worst performance in all cases in Fig. 2.

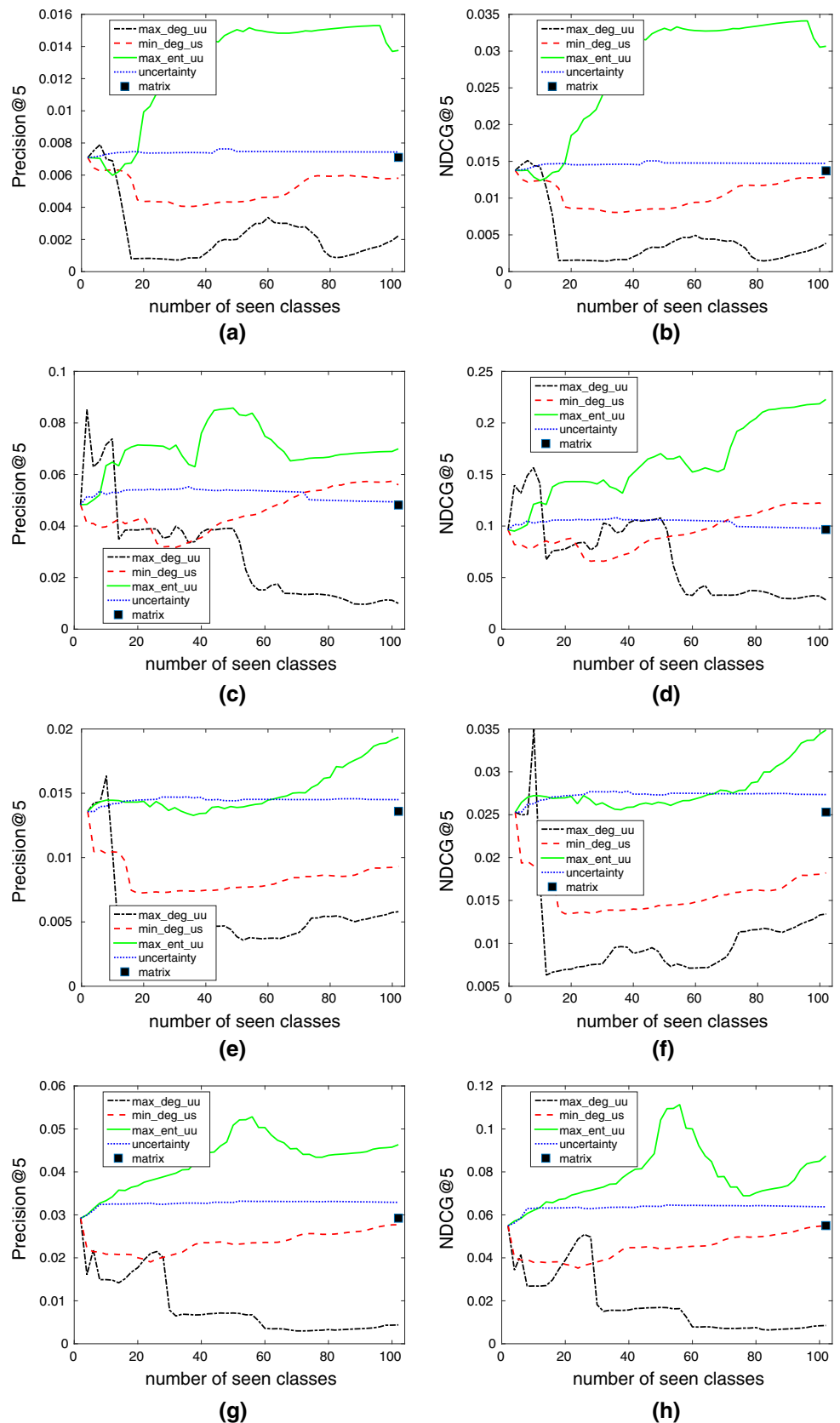
From Fig. 3a, we see that the baseline *min-deg-us* also tends to select classes that are less frequent than those selected by *max-ent-uu*, then why hasn't *min-deg-us* outperformed *max-ent-uu*? In Fig. 3b, we plot the CDF of the seen–unseen class connectivities on the same dataset, where connectivities are the row sums of  $K^{SU}$ . We see that *max-ent-uu* produces connectivities as strong as those produced by *max-deg-uu*. The classes selected by *min-deg-us* tend to have low connectivities with unseen classes, as its name suggests. The baselines *uncertainty* and *matrix* tend to produce medium such connectivities, with similar CDFs. If the connectivities are strong, then the seen classes can provide significant information about the unseen classes, and the resulting predictions are more confident. This observation also confirms our analysis of the relationship between entropy (Eq. 2) and unseen class coverage, and *max-ent-uu* will not be misled to find seen classes that have high entropies but only barely related to the unseen ones. We have similar observations on the other datasets, and we conclude that *max-ent-uu* is more likely to find seen classes that simultaneously possess the following properties: (1) discriminative about the test data (low document frequency) and (2) informative about the unseen classes (high coverage). The unique combination of these two properties helps *max-ent-uu* outperform the baselines.

## 6 Future directions

We identify a few promising research directions for XMLC.

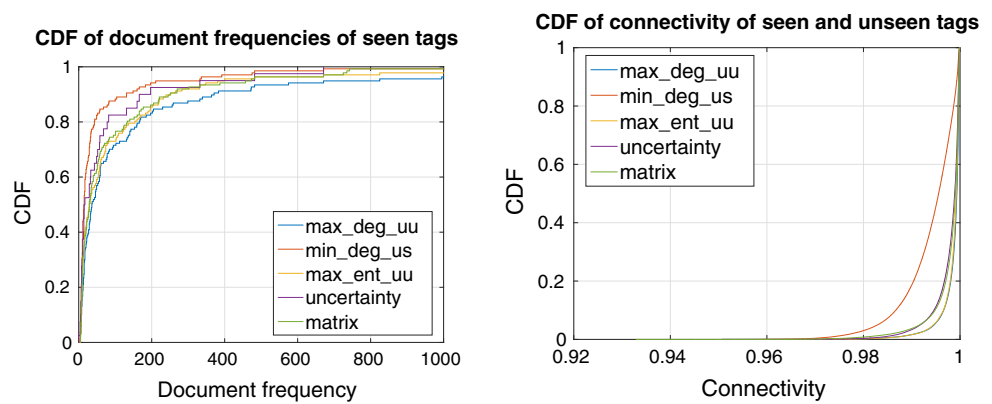
- Previous zero-shot learning algorithms utilize knowledge bases such as WordNet, web search engines, hierarchies and embedded attributes majorly through the semantic similarity information of the labels. One promising future direction is to exploit logical knowledge in the knowledge bases for better unseen label prediction. For example,

**Fig. 2** Comparisons of the proposed method and the baselines. **a** Askubuntu (Precision@5), **b** askubuntu (NDCG@5), **c** dba (Precision@5), **d** dba (NDCG@5), **e** superuser (Precision@5), **f** superuser (NDCG@5), **g** unix (Precision@5) and **h** unix (NDCG@5)





**Fig. 3** Analysis of seen–unseen tag split resulting from the proposed selection method (superuser). **a** CDF of document frequency of selected classes. **b** CDF of the seen–unseen classes connectivities



WordNet contains part–whole relations between words (e.g., “tire” is part of a “car”), and we can use such relations to find the parts given the label of the whole. Contradiction relation can also prevent the model from including labels that contradict the reliably predicted labels.

- How to choose the right knowledge base for a specific prediction task is lacking. Although previous work like [32] did empirically compare the effectiveness of different knowledge bases on their tasks, there is no formal metric defined for knowledge base selection. One important question to answer in future XMLC research is how to define such measurements based on generalization error reduction, knowledge base coverage and knowledge base consistency.
- Human-in-the-loop machine learning, such as crowdsourcing, and active learning have been proved to be critical in improving machine learning models, and XMLC is such a case. For example, when the prediction of an XMLC model is uncertain, human experts or the crowd can help resolve the uncertainty. Another way is to continuously incorporate new knowledge from human beings into the knowledge bases for better future predictions.

## 7 Conclusions

We review recent XMLC literature and categorize the published methods based on the availability of labeled data and label space compression methods. We then study active learning in the zero-shot prediction setting for the purpose of finding a small number of informative seen classes to facilitate unseen class predictions. We propose an entropy-based selection method, which is demonstrated to be able to capture the desirable distribution and strength of seen–unseen similarities. We model the similarity between classes using a beta distribution to justify the proposed entropy-based selection method. Experiments show that the proposed method

outperforms both representativeness- and uncertainty-based active learning methods.

**Acknowledgements** This work was supported in part by NSF Award III-1526499, and NVIDIA Corporation with the donation of the Titan X GPU.

## References

1. Agrawal, R. et al.: Multi-label learning with millions of labels: recommending advertiser bid phrases for web pages. In: WWW. Rio de Janeiro, Brazil (2013)
2. Balasubramanian, K., Lebanon, G.: The landmark selection method for multiple output prediction (2012)
3. Bengio, S., Weston, J., Grangier, D.: Label embedding trees for large multi-class tasks. In: NIPS (2010)
4. Bi, W., Kwok, J.T.: Multi-label classification on tree- and DAG-structured hierarchies. In: ICML. New York, NY (2011)
5. Bi, W., Kwok, J.: Efficient multi-label classification with many labels. In: ICML (2013)
6. Cesa-Bianchi, N., Gentile, C., Zaniboni, L.: Incremental algorithms for hierarchical classification. *J. Mach. Learn. Res.* **7**, 31–54 (2006)
7. Chen, Y.N., Lin, H.T.: Feature-aware label space dimension reduction for multi-label classification. In: NIPS (2012)
8. Cisse, M.M. et al.: Robust bloom filters for large multilabel classification tasks. In: NIPS (2013)
9. Cover, T.M., Thomas, J.A.: *Elements of Information Theory* (Wiley Series in Telecommunications and Signal Processing) (2006)
10. Deng, J. et al.: What does classifying more than 10,000 image categories tell us? In: ECCV (2010)
11. Deng, J. et al.: Fast and balanced: efficient label tree learning for large scale object recognition. In: NIPS (2011)
12. Dietterich, T.G., Bakiri, G.: Solving multiclass learning problems via error-correcting output codes. *J. Artif. Int. Res.* **2**, 263–286 (1995)
13. Ferng, C.-S., Lin, H.-T.: Multi-label classification with error-correcting codes. *J. Mach. Learn. Res.* **20**, 281–295 (2011)
14. Frome, A. et al.: DeViSE: a deep visual-semantic embedding model. In: NIPS. (2013)
15. Fu, Y. et al.: Transductive multi-view embedding for zero-shot recognition and annotation. In: ECCV (2014)
16. Guo, Y.: Active instance sampling via matrix partition. In: NIPS (2010)
17. Gao, T., Koller, D.: Discriminative learning of relaxed hierarchy for large-scale visual recognition. In: ICCV (2011)

18. Hsu, D.J. et al.: Multi-label prediction via compressed sensing. In: NIPS (2009)
19. Huang, K.-H., Lin, H.-T.: Cost-sensitive label embedding for multi-label classification (2016)
20. Ji, S. et al.: Extracting shared subspace for multi-label classification. In: KDD, Las Vegas, ND (2008)
21. Kapoor, A., Viswanathan, R., Jain, P.: Multilabel classification using Bayesian compressed sensing. In: NIPS (2012)
22. Lampert, C.H., Nickisch, H., Harmeling, S.: Learning to detect unseen object classes by between-class attribute transfer. In: CVPR (2009)
23. Li, X., Guo, Y.: Max-margin zero-shot learning for multi-class classification. In: AISTAT (2015)
24. Li, X. et al.: Zero-shot image tagging by hierarchical semantic embedding. In: SIGIR (2015)
25. Mensink, T., Gavves, E., Snoek, C.G.M.: COSTA: co-occurrence statistics for zero-shot classification. In: CVPR (2014)
26. Norouzi, M. et al.: Zero-shot learning by convex combination of semantic embeddings. In: CoRR (2013)
27. Palatucci, M. et al.: Zero-shot learning with semantic output codes. In: NIPS (2009)
28. Prabhu, Y., Varma, M.: FastXML: a fast, KDD, accurate and stable tree-classifier for extreme multi-label learning (2014)
29. Qi, G.-J. et al.: Two-dimensional active learning for image classification. In: CVPR (2008)
30. Rifkin, R., Klautau, A.: In defense of one-vs-all classification. *J. Mach. Learn. Res.* **5**, 101–141 (2004)
31. Robertson, S.: Understanding inverse document frequency: on theoretical arguments for IDF. *J. Doc.* **60** (2004)
32. Rohrbach, M., Stark, M., Schiele, B.: Evaluating knowledge transfer and zero-shot learning in a large-scale setting. In: CVPR (2011)
33. Rousu, J. et al.: Kernel-based learning of hierarchical multilabel classification models. *J. Mach. Learn. Res.* **7**, 1601–1626 (2006) ISSN: 1532-4435
34. Salakhutdinov, R., Mnih, A., Hinton, G.: Restricted Boltzmann machines for collaborative filtering. In: ICML (2007)
35. Socher, R. et al.: Zero-shot learning through cross-modal transfer. In: NIPS (2013)
36. Tai, F., Lin, H.-T.: Multilabel classification with principal label space transformation. *Neural Comput.* **24**(9), 2508–2542 (2012)
37. Tong, S., Koller, D.: Support vector machine active learning with applications to text classification. *J. Mach. Learn. Res.* **2**, 45–66 (2002)
38. Weston, J., Bengio, S., Usunier, N.: WSABIE: scaling up to large vocabulary image annotation. In: IJCAI (2011)
39. Weston, J., Makadia, A., Yee, H.: Label partitioning for sublinear ranking. In: ICML (2013)
40. Xu, C., Tao, D., Xu, C.: Robust Extreme Multi-label Learning. In: KDD, San Francisco, CA (2016)
41. Yu, H.-F. et al.: Large-scale multi-label learning with missing labels. In: ICML (2014)
42. Zhang, Y., Schneider, J.G.: Multi-label output codes using canonical correlation analysis. In: ICML (2011)