Efficient Multi-label Classification with Many Labels

Wei Bi WEIBI@CSE.UST.HK
James T. Kwok JAMESK@CSE.UST.HK
Department of Computer Science and Engineering, Hong Kong University of Science and Technology

Abstract

In multi-label classification, each sample can be associated with a set of class labels. When the number of labels grows to the hundreds or even thousands, existing multi-label classification methods often become computationally inefficient. In recent years, a number of remedies have been proposed. However, they are based either on simple dimension reduction techniques or involve expensive optimization problems. In this paper, we address this problem by selecting a small subset of class labels that can approximately span the original label space. This is performed by an efficient randomized sampling procedure where the sampling probability of each class label reflects its importance among all the labels. Experiments on a number of real-world multi-label data sets with many labels demonstrate the appealing performance and efficiency of the proposed algorithm.

1. Introduction

Many real-world classification problems involve multiple label classes. In multi-class classification, each sample can belong to one and only one label; whereas in multi-label classification, each sample can be associated with multiple labels. For example, in text categorization, a document can belong to the categories of "piracy", "copyright" and "software". Similarly, in bioinformatics, a gene may be associated with the functions of "transcription", "metabolism" and "protein synthesis". Image annotation is also a multi-label learning problem. Nowadays, in many social networking websites, billions of digital images, each often associated with multiple tags (e.g., "elephant", "jungle" and "africa"), are available for free download, sharing and research. For example, Flickr already has 5 billion photos and more than 20 millions unique tags as of 2010. While many of these tags may be redundant, it has been suggested that humans can still recognize between 10,000 and 100,000 unique object classes. The Dmoz data set, which is constructed by crawling webpages from the Open Directory Project, also has more than 30,000 labels. Obviously, how to handle such a large number of labels in multi-label learning is an important research problem.

A basic approach to multi-label classification is binary relevance (BR) (Tsoumakas et al., 2010), which simply trains a classifier for each label independently. In recent years, many approaches have been proposed to further improve classification performance by incorporating the label correlations (Dembczynski et al., 2010; Hariharan et al., 2010) or exploiting the label hierarchy (Bi & Kwok, 2011). However, as the number of labels (d) in many domains keeps growing, even the simple BR approach (in which the number of classifiers to train is equal to d) can easily become computationally infeasible, not to mention the more sophisticated and computationally demanding approaches.

Some recent approaches have been proposed to address the multi-label classification problem with many labels. A first attempt is by Hsu et al. (2009), who projects the d-dimensional label vector using compressed sensing, and performs training with the much lower-dimensional projected label vectors. Subsequently, many variants have been developed along this line, which use different projection mechanisms including principal component analysis (Tai & Lin, 2012), canonical correlation analysis (Zhang & Schneider, 2012) and other singular value decompositions (Chen & Lin, 2012). A common characteristic is that they all reduce the possibly large number of labels to a more manageable set of transformed labels. Yet, a major limitation is that the transformed labels, though fewer in quantity, may be more difficult to learn.

Instead of using label transformation, Balasubramanian & Lebanon (2012) proposed to train only a small

Proceedings of the 30th International Conference on Machine Learning, Atlanta, Georgia, USA, 2013. JMLR: W&CP volume 28.

subset of the labels. Since this subset come from the original labels, their learning problems will not be made more difficult. Obviously, the key issue is how to select this label subset. A good candidate should allow the non-selected labels to be faithfully and easily constructed from the selected ones. Balasubramanian & Lebanon (2012) formulated this as a a group-sparse learning problem. However, even with the recent advances in learning with structured sparsity (Bach et al., 2011), the involved optimization problem is still computationally expensive, especially when there are a lot of labels to select from.

In this paper, we alleviate this problem by proposing an efficient label selection method based on randomized sampling. Following the assumption in (Balasubramanian & Lebanon, 2012), we design the sampling probability of each label using its leverage score in the best rank-k subspace of the label matrix. Theoretical analysis shows the efficiency of this sampling scheme and its performance when coupled into the multi-label classification framework.

The rest of this paper is organized as follows. Section 2 first gives a brief review on label transformation / selection and the column subset selection problem (Drineas et al., 2006). Section 3 then presents the proposed learning algorithm. Experimental results are presented in Section 4, and the last section gives some concluding remarks.

Notation: In the sequel, n denotes the number of training samples, m is the number of features, and d is the number of labels. $\mathbf{X} \in \mathbb{R}^{n\times m}$ is the input (training) data matrix, and $\mathbf{Y} \in \{0,1\}^{n\times d}$ is the corresponding label matrix. Moreover, the transpose of vector/matrix is denoted by the superscript ^{T}, $\mathbf{A}^\dagger$ denotes the Moore-Penrose pseudo-inverse of a matrix $\mathbf{A}$, $\|\mathbf{A}\|_2$ is its spectral norm and $\|\mathbf{A}\|_F$ is the Frobenius norm, and $\mathbf{A}_{(i)}$ is the ith column of $\mathbf{A}$.

2. Related Work

2.1. Label Transformation

Hsu et al. (2009) proposed a three-step approach to address classification problems with a large number of labels. First, the high-dimensional label vector is projected to a low-dimensional space using random transformation. Next, a regression model is built for each dimension of the transformed label vector. Finally, given a test sample, the estimated label vector is projected from the low-dimensional space back to the original label space.

Recently, various improvements have been proposed. Tai & Lin (2012) found the random transformation in (Hsu et al., 2009) to be ineffective, and proposed the *principal label space transformation* (PLST) which uses principal component analysis (PCA) on the label matrix $\mathbf{Y}$. As PCA only minimizes the encoding error between $\mathbf{Y}$ and its low-dimensional representation, Chen & Lin (2012) proposed the *conditional PLST* (CPLST) that simultaneously minimizes both the encoding error and training error in the reduced-dimensional space. Similarly, Zhang & Schneider (2011) proposed to use canonical correlation analysis (CCA) that also takes both the input and output matrices into consideration. They further proposed a maximum margin formulation to learn an output coding that is predictive (based on the estimated predictions in the reduced-dimensional space) as well as discriminative (such that different label vectors have different transformed label vectors) (Zhang & Schneider, 2012). However, its optimization relies on the cutting plane algorithm (Tsochantaridis et al., 2005), which may not be efficient when there are many labels. Zhou et al. (2012) used the Gaussian random projection to form the transformed labels. During preprocessing, it extracts an auxiliary distilled label set containing the frequently appearing label subsets. However, empirically this step is expensive.

2.2. Label Selection

Recently, Balasubramanian & Lebanon (2012) proposed the *multiple output prediction landmark selection method* (MOPLMS) based on the assumption that all the output labels can be recovered by a small subset. In other words, $\mathbf{Y} \simeq \mathbf{YW}$, where $\mathbf{W} \in \mathbb{R}^{d\times d}$ is the coefficient matrix with only a few nonzero rows. Mathematically, we have

$$\min_{\mathbf{W}} \|\mathbf{Y} - \mathbf{YW}\|_F^2 + \lambda_1\|\mathbf{W}\|_{1,2} + \lambda_2\|\mathbf{W}\|_1, \qquad (1)$$

where λ_1, λ_2 are regularization parameters, $\|\mathbf{W}\|_{1,2} = \sum_{i=1}^d \sqrt{\sum_{j=1}^d W_{ij}^2}$ is the $\ell_{1,2}$ group-sparsity regularizer that encourages row sparsity of $\mathbf{W}$, and $\|\mathbf{W}\|_1 = \sum_{i,j=1}^k |W_{ij}|$ is the traditional ℓ_1-regularizer that encourages sparsity over the whole $\mathbf{W}$. However, when the number of labels d is large, $\mathbf{W}$ becomes large and problem (1) is computationally expensive. Besides, the size of the label subset cannot be explicitly controlled, and can only be indirectly varied by changing the λ_1, λ_2 parameters in (1).

2.3. Column Subset Selection Problem (CSSP)

Given a matrix $\mathbf{A} \in \mathbb{R}^{n\times d}$ and a positive integer k, the CSSP seeks to find exactly k columns of $\mathbf{A}$ so as to span $\mathbf{A}$ as much as possible. In other words, we want

to find an index set C with cardinality k such that the residual $\|\mathbf{A}-\mathbf{A}_C\mathbf{A}_C^\dagger\mathbf{A}\|_F$ is minimized. Here, $\mathbf{A}_C$ denotes the submatrix consisting of the C columns in $\mathbf{A}$, and $\mathbf{A}_C\mathbf{A}_C^\dagger$ is the projection matrix onto the k-dimensional space spanned by the columns of $\mathbf{A}_C$.

A brute-force search will need to enumerate $O(d^k)$ possible solutions. Even for a small k, this can be prohibitive when d is large. Recently, randomized sampling schemes have been proposed to find approximate solutions of the CSSP more efficiently (Drineas et al., 2006; Boutsidis et al., 2009).

A popular algorithm is the exact subspace sampling scheme (Drineas et al., 2006), which has also been incorporated by other CSSP algorithms (Boutsidis et al., 2009). For a given ϵ, it samples $O(k^2/\epsilon^2)$ columns from $\mathbf{A}$, where the probability of selecting the ith column is

$$p_i = \frac{1}{k}\|(\mathbf{V}_{\mathbf{A},k}^T)_{(i)}\|_2^2, \tag{2}$$

and $\mathbf{V}_{\mathbf{A},k}$ is the matrix containing the top k right singular vectors of $\mathbf{A}$. Intuitively, p_i corresponds to the leverage score of $\mathbf{A}_{(i)}$ on the best rank-k subspace of $\mathbf{A}$ (Boutsidis et al., 2009). In statistical diagnostic regression analysis, leverage scores can be used to measure outliers. Thus, (2) provides a bias toward columns that are outlying, which play an important role in spanning the subspace. With probability $1-\frac{1}{e}$, the approximation error of the resultant approximation $\|\mathbf{A}-\mathbf{A}_C\mathbf{A}_C^\dagger\mathbf{A}\|_F$ is upper-bounded by $(1+\epsilon)\|\mathbf{A}-\mathbf{A}_k\|_F$, where $\mathbf{A}_k$ is the best rank-k approximation of $\mathbf{A}$.

Recently, Boutsidis et al. (2009) proposed to first sample $\Theta(k\log k)$ columns from $\mathbf{A}$ with the probabilities in (2), and then perform the rank-revealing QR (RRQR) decomposition (Gu & Eisenstat, 1996) on a scaled version of the sampled columns of $\mathbf{V}_{\mathbf{A},k}^T$. This allows the extraction of exactly k columns from $\mathbf{A}$. With probability 0.8, the error bound is $\|\mathbf{A}-\mathbf{A}_C\mathbf{A}_C^\dagger\mathbf{A}\|_F \le \Theta(k\log^{\frac{1}{2}}k)\|\mathbf{A}-\mathbf{A}_k\|_F$.

3. Proposed Algorithm

Recall that CSSP aims to find k columns from a given matrix $\mathbf{A}$ so that the reconstruction error is minimized. Obviously, this shares the same goal as MOPLMS (Balasubramanian & Lebanon, 2012), with the label matrix $\mathbf{Y}$ playing the role of $\mathbf{A}$, i.e.,

$$\min_C \|\mathbf{Y}-\mathbf{Y}_C\mathbf{Y}_C^\dagger\mathbf{Y}\|_F. \tag{3}$$

While MOPLMS relies on an expensive optimization problem to select the labels (columns), here we will use an efficient CSSP variant (Section 3.1). Once this subset of k labels are selected, a binary classifier is trained for each of them (alternatively, the k labels can be learned jointly as in multi-task learning). In total, k binary classifiers are needed. In contrast, a direct application of BR requires the training of d binary classifiers, where $d \gg k$.

As label selection is now considered as a CSSP, in principle one can use any of the algorithms in Section 2. However, this may not be entirely satisfactory. While we aim at selecting exactly k columns of $\mathbf{Y}$, the algorithm in (Drineas et al., 2006) selects a lot more than k columns. As for the algorithm in (Boutsidis et al., 2009), though it can output exactly k columns, it needs to first select $c=\Theta(k\log k)$ columns. Empirically, a proper choice of c can be sensitive to the label matrix (Mahoney, 2011). Moreover, performing the RRQR decomposition in its deterministic step takes $O(c^2k\log\sqrt{c})$ time (Boutsidis et al., 2009). As will be demonstrated in Section 4.3, this can be even more computationally expensive than the sampling step itself.

In Section 3.1, we propose a novel variant of (Drineas et al., 2006; Boutsidis et al., 2009) which directly selects k different columns in $\mathbf{Y}$, while ensuring efficiency and a good approximation error bound. Section 3.2 discusses the prediction mechanism. Section 3.3 provides an error analysis, and Section 3.4 discusses how this can be extended to the use of kernels.

3.1. Proposed CSSP Randomized Sampling

The proposed procedure is shown in Algorithm 1. As in (Drineas et al., 2006; Boutsidis et al., 2009), we first perform partial SVD on $\mathbf{Y}$ and pick the top k right singular vectors $\mathbf{V}_{\mathbf{Y},k}\in\mathbb{R}^{m\times k}$. For notational simplicity, we will use $\mathbf{V}_k$ for $\mathbf{V}_{\mathbf{Y},k}$ in the sequel. The columns in $\mathbf{Y}$ are sampled with replacement, with the probability for selecting the ith column being

$$p_i = \frac{1}{k}\|(\mathbf{V}_k^T)_{(i)}\|_2^2, \tag{4}$$

as in (2). However, instead of performing a fixed number of sampling trials, we continue sampling until k different columns are selected (steps 4-9).

Similar to (Boutsidis et al., 2009), the following Proposition shows that the $(\mathbf{V}_k^T)_C$ matrix sampled is full rank with high probability.

Proposition 1. *If the* WHILE *loop stops in T trials, where*

$$T = \frac{2c_0^2k}{\epsilon^2}\log\frac{c_0^2k}{\epsilon^2} \tag{5}$$

and c_0 is a constant in Theorem 3.1 of (Rudelson &

Algorithm 1 Multi-label classification via CSSP (ML-CSSP).

1: Compute $\mathbf{V}_k$, the top k right singular vectors of $\mathbf{Y}$.
2: Compute the sampling probability p_i for each column in $\mathbf{Y}$ using (4).
3: $C \leftarrow \emptyset$.
4: **while** $|C| < k$ **do**
5: Select an integer from $\{1, 2, \ldots, m\}$ where the probability of selecting i is equal to p_i.
6: **if** $i \notin C$ **then**
7: $C \leftarrow C \cup \{i\}$.
8: **end if**
9: **end while**
10: Train the classifier $f(\mathbf{x})$ from $\{\mathbf{x}^{(n)}, \mathbf{y}_C^{(n)}\}_{n=1}^N$.
11: Given a new test point $\mathbf{x}$, obtain its prediction $\mathbf{h}$ using $f(\mathbf{x})$ and return $\hat{\mathbf{y}}$ by rounding $\mathbf{h}^T \mathbf{Y}_C^\dagger \mathbf{Y}$.

Vershynin, 2007), then $(\mathbf{V}_k^T)_C$ *is full rank with probability* $1 - 4\epsilon^2$.

Moreover, one can obtain the following error bound, which shows that the obtained approximation error is close to that based on the best rank-k approximation, with a factor of $\Theta(1)$.

Corollary 1. *With probability* $0.9 - 4\epsilon^2$, $\|\mathbf{Y} - \mathbf{Y}_C \mathbf{Y}_C^\dagger \mathbf{Y}\|_F \leq \Theta(1)\|\mathbf{Y} - \mathbf{Y}_k\|_F$, *where* $\mathbf{Y}_k$ *is the best rank-k approximation of* $\mathbf{Y}$.

The following Proposition shows that, with probability at least 0.9, k different columns will be selected in $O(k \log k)$ trials. Thus, in the worse case, we do not need to sample more columns than the algorithm in (Boutsidis et al., 2009).

Proposition 2. *With probability at least 0.9, k different columns are selected in $O(k \log k)$ sampling trials.*

Though the above results suggest that $O(k \log k)$ columns may still need to be sampled in the worst case, empirical results in Section 4.3 show that the T obtained is much smaller than $k \log k$.

3.1.1. COMPARISON WITH OTHER METHODS

Table 1 compares the proposed algorithm with the existing CSSP algorithms in (Drineas et al., 2006) and (Boutsidis et al., 2009). To compute the sampling probabilities, all three CSSP-based algorithms have to first obtain the top k singular vectors of the label matrix $\mathbf{Y}$. This takes $O(\min\{nd^2, n^2 d\})$ time in general, but can be reduced to $O(ndk)$ time by using Lanzcos/Arnoldi algorithms as $\mathbf{Y}$ is typically sparse. The second term in the time complexity comes from the number of sampling trials (which is $O(k^2/\epsilon^2)$ for (Drineas et al., 2006), $O(k \log k)$ for (Boutsidis et al., 2009) and ours). Finally, the algorithm in (Boutsidis et al., 2009) involves an additional RRQR decomposition after the sampling step, which takes $O(c^2 k \log \sqrt{c}) = O(k^3 \log^2 k \log(k \log k))$ time for $c = \Theta(k \log k)$.

For easy comparison, we also show the time complexities for PLST and CPLST in Table 1. PLST is fast, as it only requires a single SVD on the label matrix, which takes $O(ndk)$ time. CPLST needs to compute the pseudoinverse of the input matrix $\mathbf{X}$ (which is as expensive as computing its SVD) and a SVD on the matrix $\mathbf{Y}^T \mathbf{X} \mathbf{X}^\dagger \mathbf{Y}$. In general, these two matrices are dense, and so CPLST takes $O(\min\{nm^2, n^2 m\} + d^3)$ time for these two operations, and is much more expensive. The time complexity of MOPLMS cannot be directly compared as it involves numerical optimization. Empirically, as will be demonstrated in Section 4.2, this is much more expensive.

3.2. Prediction

On prediction, we first apply the k learned classifiers on a new test sample to obtain its k-dimensional prediction vector $\mathbf{h}$. Note from (3) that $\mathbf{Y} \simeq \mathbf{Y}_C \mathbf{Y}_C^\dagger \mathbf{Y}$. Each row of $\mathbf{Y}$ (which corresponds to the d labels of a particular sample) can thus be approximated as the product of the corresponding row in $\mathbf{Y}_C$ (which corresponds to the k selected labels of the same sample) with $\mathbf{Y}_C^\dagger \mathbf{Y}$. Given $\mathbf{h}$, a d-dimensional label vector $\hat{\mathbf{y}}$ can be recovered as $\mathbf{h}^T \mathbf{Y}_C^\dagger \mathbf{Y}$. This is further rounded to produce a binary classification output.

3.3. Error Analysis

In this section, we analyze the root mean square error (RMSE) on the training examples, which is defined as

$$\text{RMSE} \equiv \frac{1}{\sqrt{n}} \|\hat{\mathbf{Y}} - \mathbf{Y}\|_F, \tag{6}$$

where $\hat{\mathbf{Y}}$ is the estimated label matrix. As $\hat{\mathbf{Y}}, \mathbf{Y}$ are binary, the squared RMSE is also proportional to the commonly used Hamming loss $\frac{1}{nd}\|\mathbf{Y} - \hat{\mathbf{Y}}\|^2$.

Denote the estimated label matrix using $f(\mathbf{x})$ in the selected label dimensions as $\mathbf{H}$. Let $\mathbf{Y}_k$ be the best rank-k approximation of $\mathbf{Y}$. The following Proposition bounds the training RMSE for Algorithm 1.

Proposition 3. *With the conditions in Proposition 1, we have, with probability* $1 - 4\epsilon^2$,

$$\textit{RMSE} \leq \frac{2}{\sqrt{n}} \left(\|\mathbf{H} - \mathbf{Y}_C\|_F + \|\mathbf{Y} - \mathbf{Y}_C \mathbf{Y}_C^\dagger \mathbf{Y}\|_F \right). \tag{7}$$

Proof. Let $\hat{\mathbf{Y}} = \text{round}(\mathbf{H} \mathbf{Y}_C^\dagger \mathbf{Y})$ be the reconstructed

Table 1. Comparison of the various CSSP sampling algorithms, together with PLST and CPLST.

	(Drineas et al., 2006)	(Boutsidis et al., 2009)	ours	PLST	CPLST
time complexity	$O(ndk)$ $+O(\frac{k^2}{\epsilon^2})$	$O(ndk)+O(k\log k)$ $+O(k^3\log^2 k\log(k\log k))$	$O(ndk)$ $+O(k\log k)$	$O(ndk)$	$O(\min\{nm^2,n^2m\})$ $+O(d^3)$
#sampling trials	$\Theta(\frac{k^2}{\epsilon^2})$	$\Theta(k\log k)$	$O(k\log k)$	-	-
approximation ratio	$1+\epsilon$	$\Theta(k\log^{1/2} k)$	$\Theta(1)$	-	-
probability it holds	1-1/e	0.8	$0.9-4\epsilon^2$	-	-

label vector after rounding, where round$(\cdot)$ rounds each element of the matrix argument to the nearest 0 or 1. Since both $\hat{\mathbf{Y}}$ and $\mathbf{Y}$ are binary, $(\hat{\mathbf{Y}}-\mathbf{Y})_{ij}$ is either 1 or 0.

1. $(\hat{\mathbf{Y}}-\mathbf{Y})_{ij}=1$: This happens iff $(\mathbf{H}\mathbf{Y}_C^\dagger\mathbf{Y})_{ij}>\frac{1}{2}$ when $\mathbf{Y}_{ij}=0$; and $<\frac{1}{2}$ otherwise. In both cases, $(\mathbf{H}\mathbf{Y}_C^\dagger\mathbf{Y}-\mathbf{Y})_{ij}^2\geq(\frac{1}{2})^2=\frac{1}{4}$. Since $(\hat{\mathbf{Y}}-\mathbf{Y})_{ij}=1$, we can also write it as
$$(\mathbf{H}\mathbf{Y}_C^\dagger\mathbf{Y}-\mathbf{Y})_{ij}^2\geq\frac{1}{4}(\hat{\mathbf{Y}}-\mathbf{Y})_{ij}^2. \quad (8)$$
2. $(\hat{\mathbf{Y}}-\mathbf{Y})_{ij}=0$: In this case, (8) also holds trivially.

Thus,
$$\|\hat{\mathbf{Y}}-\mathbf{Y}\|_F=\sqrt{\sum_{ij}(\hat{\mathbf{Y}}-\mathbf{Y})_{ij}^2}\leq 2\|\mathbf{H}\mathbf{Y}_C^\dagger\mathbf{Y}-\mathbf{Y}\|_F. \quad (9)$$

Using the triangle inequality for norms and the fact that $\|\mathbf{AB}\|_F\leq\|\mathbf{A}\|_F\|\mathbf{B}\|_2$, we have
$$\begin{aligned}&\|\mathbf{H}\mathbf{Y}_C^\dagger\mathbf{Y}-\mathbf{Y}\|_F\\&=\|(\mathbf{H}-\mathbf{Y}_C)\mathbf{Y}_C^\dagger\mathbf{Y}+\mathbf{Y}_C\mathbf{Y}_C^\dagger\mathbf{Y}-\mathbf{Y}\|_F\\&\leq\|\mathbf{H}-\mathbf{Y}_C\|_F\|\mathbf{Y}_C^\dagger\mathbf{Y}\|_2+\|\mathbf{Y}_C\mathbf{Y}_C^\dagger\mathbf{Y}-\mathbf{Y}\|_F \quad (10)\end{aligned}$$

From Proposition 1, $(\mathbf{V}_k^T)_C$ is full rank (and thus has rank k) with probability $1-4\epsilon^2$. As $(\mathbf{V}_k^T)_C$ is a submatrix of $(\mathbf{V}^T)_C$, $\text{rank}((\mathbf{V}^T)_C)\geq\text{rank}((\mathbf{V}_k^T)_C)$. Since $(\mathbf{V}^T)_C$ has k different columns, $\text{rank}((\mathbf{V}^T)_C)\leq k$. Let the SVD of $\mathbf{Y}$ be $\mathbf{Y}=\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$. Then $\mathbf{Y}_C=\mathbf{U}\mathbf{\Sigma}(\mathbf{V}^T)_C$, which is full rank with rank k. Assuming that $\mathbf{V}^T$ is of rank d, we have $\mathbf{Y}_C^\dagger=((\mathbf{V}^T)_C)^T\mathbf{\Sigma}^{-1}\mathbf{U}^T$ as $\mathbf{Y}_C\mathbf{Y}_C^\dagger\mathbf{Y}_C=\mathbf{Y}_C$ and $\mathbf{Y}_C^\dagger\mathbf{Y}_C\mathbf{Y}_C^\dagger=\mathbf{Y}_C^\dagger$. Thus,
$$\begin{aligned}\|\mathbf{Y}_C^\dagger\mathbf{Y}\|_2&=\|(\mathbf{U}\mathbf{\Sigma}(\mathbf{V}^T)_C)^\dagger\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T\|_2\\&=\|((\mathbf{V}^T)_C)^T\mathbf{\Sigma}^{-1}\mathbf{U}^T\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T\|_2\\&=\|(\mathbf{V}^T)_C)^T\mathbf{V}^T\|_2=1. \quad (11)\end{aligned}$$

Result follows by combining (9), (10), (11). □

The error bound in (7) consists of two parts. Intuitively, the first term $\|\mathbf{H}-\mathbf{Y}_C\|_F$ represents the training error between the learned label submatrix and the target label submatrix $\mathbf{Y}_C$ selected by the algorithm; while the second term $\|\mathbf{Y}-\mathbf{Y}_C\mathbf{Y}_C^\dagger\mathbf{Y}\|_F$ represents the encoding error in projecting the full label matrix $\mathbf{Y}$ onto the selected labels $\mathbf{Y}_C$.

Combining with Corollary 1, we immediately obtain the following.

Corollary 2. *With the conditions in Proposition 1, we have, with probability* $0.9-4\epsilon^2$*, that*
$$RMSE\leq\frac{2}{\sqrt{n}}\|\mathbf{H}-\mathbf{Y}_C\|_F+\frac{2}{\sqrt{n}}\Theta(1)\|\mathbf{Y}-\mathbf{Y}\mathbf{V}_k\mathbf{V}_k^T\|_F.$$

This is similar to the error bound for PLST in (Tai & Lin, 2012), namely, RMSE $\leq\frac{2}{\sqrt{n}}\|\mathbf{H}-\mathbf{Y}\mathbf{V}_k\|_F+\frac{2}{\sqrt{n}}\|\mathbf{Y}-\mathbf{Y}\mathbf{V}_k\mathbf{V}_k^T\|_F$, where $\|\mathbf{H}-\mathbf{Y}\mathbf{V}_k\|_F$ is the training error between the learned label submatrix and the projected label matrix $\mathbf{Y}\mathbf{V}_k$.

3.4. Kernel Extension

Instead of assuming that the columns of $\mathbf{Y}$ are spanned by a small subset of its columns, we can first map the columns to some kernel-induced feature space before taking the approximation. In other words, we try to minimize $\|\tilde{\mathbf{Y}}-\tilde{\mathbf{Y}}_C\tilde{\mathbf{Y}}_C^\dagger\tilde{\mathbf{Y}}\|_F$, where $\tilde{\mathbf{Y}}$ is the mapped label matrix.

As in Section 3.1, the probability for selecting the ith column can be similarly defined as $p_i=\frac{1}{k}\|(\tilde{\mathbf{V}}_k^T)_{(i)}\|_2^2$, where $\tilde{\mathbf{Y}}=\tilde{\mathbf{U}}\tilde{\mathbf{\Sigma}}\tilde{\mathbf{V}}^T$. It can be easily seen that the $\tilde{\mathbf{V}}_k$'s are the same as the top k eigenvectors of the kernel matrix $\mathbf{K}=\tilde{\mathbf{Y}}^T\tilde{\mathbf{Y}}$, as $\mathbf{K}=\tilde{\mathbf{Y}}^T\tilde{\mathbf{Y}}=\tilde{\mathbf{V}}\tilde{\mathbf{\Sigma}}\tilde{\mathbf{U}}^T\tilde{\mathbf{U}}\tilde{\mathbf{\Sigma}}\tilde{\mathbf{V}}^T=\tilde{\mathbf{V}}\tilde{\mathbf{\Sigma}}^2\tilde{\mathbf{V}}^T$.

On testing, the prediction $\hat{\mathbf{y}}$ of a pattern $\mathbf{x}$ can be obtained by directly minimizing $\|\tilde{\mathbf{y}}-\tilde{\mathbf{h}}^T\tilde{\mathbf{Y}}_C^\dagger\tilde{\mathbf{Y}}\|_F$, where $\tilde{\mathbf{y}}$ is the kernel-mapped vector of $\hat{\mathbf{y}}$. For the RBF kernel, it can be shown that the optimal $\hat{\mathbf{y}}$ can be easily obtained in a component-by-component manner.

4. Experiments

In this section, we perform experiments on a number of benchmark real-world data sets[1] (Table 2).

- cal500 (Turnbull et al., 2008): It contains songs by different artists. Each song is annotated by a vocabulary of 174 tags representing genres, instruments, emotions, and other related concepts.
- corel5k (Duygulu et al., 2002): It contains images from the Stock Photo CDs. Each image is anno-

[1]Downloaded from http://mulan.sourceforge.net and http://lshtc.iit.demokritos.gr/LSHTC2_datasets

tated with 1 to 5 keywords. In total, there are 374 keywords.

- delicious (Tsoumakas et al., 2008): It contains the text data of web pages along with their tags from the del.icio.us social bookmarking site.
- Eur-Lex (Mencía & Fürnkranz, 2008): It contains a collection of documents on the European Union law. The labels include several EuroVoc descriptors, directory codes and subject matters. Here, we use the first two, as they have more labels.
- dmoz: It is constructed by crawling webpages from the Open Directory Project, and used in the Second Pascal Challenge on Large Scale Hierarchical Text classification. Here, we include both the internal and leaf classes.

Table 2. Data sets used in the experiment.

data set	#samples	#features	#labels
cal500	502	68	174
corel5k	5,000	499	374
delicious	16,105	500	983
EUR-Lex (dc)	19,348	5,000	412
EUR-Lex (desc)	19,348	5,000	3,993
dmoz	394,756	829,208	35,437

Using linear regression as the learner, the proposed ML-CSSP and its kernel version ML-CSSP-Knl (with the RBF kernel) are compared with various recent multi-label output coding methods: (i) PLST (Tai & Lin, 2012); (ii) CPLST (Chen & Lin, 2012); (iii) MOPLMS (Balasubramanian & Lebanon, 2012); (iv) compressed labeling (CL) (Zhou et al., 2012). We also compare with the standard baseline of binary relevance (BR), which trains a classifier for each label independently. All the methods are implemented in MatLab. We do not compare with the compressed-sensing-based method in (Hsu et al., 2009), as its performance is already shown to be inferior to PLST (Tai & Lin, 2012).

The number of selected (or transformed) labels is set to $k = 0.1d$, except for dmoz in which we use $k = 300$ (i.e., $k \simeq 0.01d$). Note that MOPLMS cannot set k explicitly. Thus, we try different settings of its regularization parameter, and pick the one whose resultant k is closest to $0.1d$.

For performance evaluation, we will use two popular measures: (i) the RMSE as defined in (6); and (ii) the micro-averaged area under the precision-recall curve (AUPRC) (Vens et al., 2008). Following (Mencía & Fürnkranz, 2008), we perform 10-fold cross-validation, except on the large dmoz data set (for which we randomly select 40,000 samples for training and the rest for testing, and repeat 3 times). All experiments are run on a PC with quad-core 3.40 GHz Intel i7-3770 CPU and 32 GB RAM.

4.1. Accuracy

Table 3 shows the RMSE results obtained. Recall that MOPLMS is very computationally expensive, and so cannot be run on most of the larger data sets. For the largest dmoz data set, ML-CSSP-Knl also runs out of memory as it has to perform a partial SVD on the dense 35K $\times$ 35K kernel matrix. Similarly, CPLST fails as it has to compute the pseudoinverse of the 350K $\times$ 800K input data matrix. For CL, computing the distilled label set in its preprocessing step already takes more than 72 hours. Moreover,

- The performance of CPLST is comparable with that of PLST. This also agrees with the empirical results in (Chen & Lin, 2012), and suggests that the input data matrix may provide only little information for label transformation or selection.
- CL does not perform well in our experiments. Empirically, it is sensitive to the settings of a number of parameters, such as the number of label clusters, the threshold in the underlying spectral clustering algorithm, etc.
- ML-CSSP-Knl is better than ML-CSSP on the multimedia data sets (cal500 and corel5k), but does not outperform on the text data sets. This agrees with the common observation that the linear kernel is often sufficient for the text data, while the nonlinear kernel is more beneficial on non-text, lower-dimensional data.

Overall, ML-CSSP and ML-CSSP-Knl achieve the best accuracy on five of the six data sets.

Table 4 shows the AUPRC results,[2] which is obtained by varying the rounding threshold for each of the methods from 0 to 1. As can be seen, ML-CSSP outperforms the others on 5 of the 6 data sets. Note that PLST sometimes performs much worse than the other methods (e.g., on the data sets of delicious, EUR-Lex (desc) and dmoz).

4.2. Time

In this section, we compare the time performance of the various multi-label output coding methods. With our experimental setup, they all have the same number of learning tasks after label transformation/selection. Hence, we will only compare their encoding time, i.e., the time to perform label transformation/selection. Results are shown in Table 5. As can be seen,

[2] ML-CSSP-Knl obtains the binary prediction outputs directly without requiring a threshold, and so its AUPRC results are not reported.

Table 3. Testing RMSE's obtained on the various data sets (number in square brackets indicates the rank). Methods that cannot be run are denoted "-". The best and comparable results (according to the pairwise t-test with 95% confidence) are highlighted.

	(label selection methods)			(label transformation methods)			baseline
data set	ML-CSSP	ML-CSSP-Knl	MOPLMS	PLST	CPLST	CL	BR
cal500	4.93 ± 0.10 [2]	**4.89** ± 0.11 [1]	5.04 ±0.10 [5]	4.97 ± 0.12 [3]	5.00 ± 0.11 [4]	5.70 ± 0.39 [7]	5.06 ± 0.11 [6]
corel5k	**1.89** ± 0.02 [1]	**1.87** ±0.02 [1]	**1.89** ± 0.05 [1]	1.91 ± 0.03 [4]	1.91 ± 0.02 [4]	2.71± 0.22 [7]	1.91 ± 0.03 [4]
delicious	4.29 ± 0.02 [4]	4.36 ± 0.02 [5]	-	4.27 ± 0.02 [3]	**4.26** ± 0.02 [1]	5.58 ± 0.18 [6]	**4.26** ± 0.01 [1]
EUR-Lex (dc)	**1.22** ± 0.03 [1]	1.52 ± 0.11 [5]	-	**1.22** ± 0.03 [1]	**1.23** ± 0.03 [1]	2.03 ± 0.04 [6]	1.50 ± 0.07 [4]
EUR-Lex (desc)	**2.93** ± 0.09 [1]	3.88 ± 0.25 [5]	-	3.02 ± 0.10 [2]	3.06 ± 0.11 [3]	4.51 ± 0.18 [6]	3.51± 0.18 [4]
dmoz	**2.83** ± 0.01 [1]	-	-	2.95 ± 0.02 [2]	-	-	4.02 ± 0.03 [3]

Table 4. AUPRC's obtained on the various data sets. The best and comparable results (according to the pairwise t-test with 95% confidence) are highlighted.

data set	ML-CSSP	MOPLMS	PLST	CPLST	CL	BR
cal500	**0.500** ± 0.031 [1]	0.459 ± 0.030 [3]	0.488 ± 0.035 [2]	0.412 ± 0.035 [5]	0.169 ± 0.028 [6]	0.442 ± 0.034 [4]
corel5k	**0.089** ± 0.031 [1]	0.080 ± 0.029 [4]	0.079 ± 0.029 [5]	0.082 ± 0.029 [3]	0.011 ± 0.004 [6]	0.083 ± 0.029 [2]
delicious	0.220 ± 0.005 [3]	-	0.182 ± 0.045 [4]	0.227 ± 0.005 [2]	0.089 ± 0.007 [5]	**0.237** ± 0.005 [1]
EUR-Lex (dc)	**0.180** ± 0.013 [1]	-	**0.180** ± 0.031 [1]	0.167 ± 0.012 [4]	0.036 ± 0.001 [5]	0.173 ± 0.015 [3]
EUR-Lex (desc)	**0.094** ± 0.008 [1]	-	0.018 ± 0.003 [4]	0.086 ± 0.009 [2]	0.016 ± 0.006 [5]	0.086 ± 0.009 [2]
dmoz	**0.016** ± 0.000 [1]	-	0.001 ± 0.000 [3]	-	-	0.012 ± 0.000 [2]

- CL has the shortest encoding time as it uses random label transformation. However, it has to first obtain a distilled label set as preprocessing. Taking this also into account, CL becomes the slowest.

- On encoding, both ML-CSSP and PLST have to perform SVD, but CSSP needs an additional $O(k \log k)$ time for the randomized sampling step (Table 1). Thus, CSSP is always slower than PLST. However, when both the number of samples (n) and the number of labels (d) are large, this extra $O(k \log k)$ time becomes less significant in comparison with the $O(ndk)$ time for the SVD.

- CPLST has to compute the pseudoinverse of the input data matrix, which takes $O(\min\{nm^2, n^2m\})$ time. Hence, it can be slow when this matrix is large (as on the EUR-Lex (dc) and EUR-Lex (desc) data sets).

- ML-CSSP-Knl is much slower than CSSP, as it needs to compute the $d \times d$ kernel matrix on the labels. Moreover, since the kernel matrix is dense, its SVD can be much slower than SVD on the sparse label matrix.

- MOPLMS has the longest encoding time as it needs to solve a complex optimization problem.

Overall, the encoding speeds of ML-CSSP and PLST are reasonably fast. Both of them take less than half an hour even on the largest data set of dmoz. Moreover, they reduce the number of learning tasks by 99% when compared to BR. In combination with the accuracy comparison in Section 4.1, we can conclude that ML-CSSP is both fast and accurate.

4.3. Comparison with (Boutsidis et al., 2009)

Here, we compare the proposed method with the CSSP algorithm in (Boutsidis et al., 2009). As suggested in (Boutsidis et al., 2009), we set the number of sampled columns to $s = 2k \log k$, and the RRQR decomposition is then used to extract exactly k columns (labels) from these s columns.

Table 6 shows the number of sampling trials and encoding time. As can be seen, the proposed method always samples a much smaller number of columns, and its encoding is also faster. Recall that the RRQR decomposition in (Boutsidis et al., 2009) operates on the matrix $(\mathbf{V}_k^T)_S$, where S is the subset of indices selected in the sampling step (with $|S| = s$). This matrix is of size $k \times s = k \times 2k \log k$, which is around 400 × 5K for EUR-Lex (desc) and 300 × 3.5K for dmoz. These are too large for the RRQR algorithm, which cannot finish in an hour. Thus, the CSSP algorithm in (Boutsidis et al., 2009) is not scalable enough for problems with a large number of labels. Table 7 compares the two methods in terms of the approximation ratio $\frac{\|\mathbf{Y}-\mathbf{Y}_C\mathbf{Y}_C^\dagger\mathbf{Y}\|_F}{\|\mathbf{Y}-\mathbf{Y}_k\|_F}$ and testing RMSE. As can be seen, both methods are comparable.

Overall, both algorithms have similar accuracy but the proposed method is more efficient and can be used on problems with much larger number of labels.

We further test the probability of $(\mathbf{V}_k^T)_C$ to be full rank in Proposition 1. Table 8 shows that using the sampling probabilities in (4), the $(\mathbf{V}_k^T)_C$ obtained is always full rank in all the repetitions. In contrast, if we use the more naive uniform sampling scheme, the probabilities of obtaining a full-rank $(\mathbf{V}_k^T)_C$ drop significantly. Thus, empirically, our sampling can obtain

Table 5. Encoding time (in seconds) on the various data sets (number in square brackets indicates the rank). For CL, its preprocessing time is shown in parentheses. The best and comparable results (according to the pairwise t-test with 95% confidence) are highlighted.

	(label selection methods)			(label transformation methods)		
data set	ML-CSSP	ML-CSSP-Knl	MOPLMS	PLST	CPLST	CL
cal500	0.0 ± 0.0 [2]	0.3 ± 0.0 [4]	10.7 ± 1.7 [5]	**0.0** ± 0.0 [1]	0.0 ± 0.0 [3]	(182+) 0.0 ± 0.0 [6]
corel5k	0.2 ± 0.0 [2]	17.2 ± 0.8[4]	36.8 ± 6.8 [5]	**0.2** ± 0.0 [1]	0.3 ± 0.0 [3]	(292+) 0.0 ± 0.0 [6]
delicious	11.6 ± 2.9 [2]	133.7 ± 2.5 [4]	-	**11.6** ± 2.9 [1]	16.0 ± 3.5 [3]	(5675+) 0.2 ± 0.0 [5]
EUR-Lex (dc)	4.0 ± 0.3 [2]	17.93 ± 0.90 [4]	-	**4.0** ± 0.3 [1]	352.9 ± 32.3 [3]	(547+) 0.1 ± 0.0 [5]
EUR-Lex (desc)	153.8 ± 20.0 [2]	1839.3 ± 202.3 [4]	-	**153.8** ± 20.0 [1]	511.7 ± 60.1 [3]	(15582+) 3.5 ± 0.2 [5]
dmoz	1428.9 ± 59.2 [2]	-	-	**1428.7** ± 59.2 [1]	-	-

Table 6. Number of sampling trials and encoding time for the proposed CSSP algorithm and the algorithm in (Boutsidis et al., 2009) (which is denoted as BMD09).

	sampling trials		encoding time (sec)	
data set	ML-CSSP	BMD09	ML-CSSP	BMD09
cal500	18 ± 1	99 ± 0	0.0 ± 0.00	0.1 ± 0.0
corel5k	57 ± 7	271 ± 0	0.2 ± 0.0	0.8 ± 0.2
delicious	134 ± 7	902 ± 0	11.6 ± 2.9	165.6 ± 10.8
EUR-Lex (dc)	76 ± 11	297 ± 0	4.03 ± 0.3	4.4 ± 0.5
EUR-Lex (desc)	660 ± 22	-	153.8 ± 20.0	-
dmoz	472 ± 5	-	1428.9 ± 59.2	-

Table 7. Approximation ratio and testing RMSE for the proposed CSSP algorithm and the algorithm in (Boutsidis et al., 2009) (which is denoted as BMD09).

	approximation ratio		testing RMSE	
data set	ML-CSSP	BMD09	ML-CSSP	BMD09
cal500	1.33 ± 0.02	1.24 ± 0.04	4.93 ± 0.10	4.97 ± 0.09
corel5k	1.28 ± 0.02	1.33 ± 0.04	1.89 ± 0.02	1.90 ± 0.03
delicious	1.15 ± 0.01	1.54 ± 0.01	4.29 ± 0.02	4.34 ± 0.02
EUR-Lex (dc)	1.03 ± 0.01	1.35 ± 0.02	1.22 ± 0.03	1.21 ± 0.01

full rank $(\mathbf{V}_k^T)_C$ with high probability, thus Corollary 1 holds with high probability.

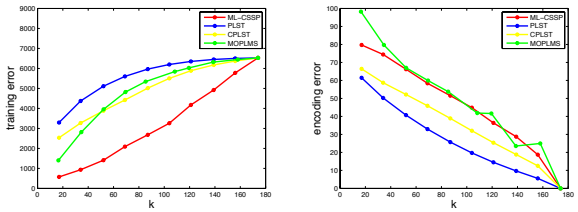
Table 8. Probabilities that $(\mathbf{V}_k^T)_C$ is full rank in the 10 folds of 10-fold cross-validation (for dmoz, it is over the 3 repetitions used).

sampling method	cal500	corel5k	delicious	EUR-Lex (dc)	EUR-Lex (desc)	dmoz
proposed	100%	100%	100%	100%	100%	100%
uniform	100%	50%	100%	0%	0%	0%

4.4. Variation with the Number of Selected Labels

In this section, we demonstrate the tradeoff between training error and encoding error as discussed in Section 3.3. The number of selected labels k is varied from $0.1d$ to d on the smallest data sets cal500. For each k, we show the training error ($\|\mathbf{H} - \mathbf{Y}_C\|_F$ for ML-CSSP and MOPLMS, and $\|\mathbf{H} - \mathbf{Y}\mathbf{V}_k\|_F$ for PLST and CPLST) and the encoding error ($\|\mathbf{Y} - \mathbf{Y}_C\mathbf{Y}_C^\dagger\mathbf{Y}\|_F$ for ML-CSSP, $\|\mathbf{Y} - \mathbf{Y}\mathbf{W}\|_F$ for MOPLMS, and $\|\mathbf{Y} - \mathbf{Y}_C\mathbf{V}_k\mathbf{V}_k^T\|_F$ for PLST and CPLST).

Figure 1 shows the results. As can be seen, the training error increases with k, as there are more tasks to be learned. Moreover, ML-CSSP and MOPLMS have lower training errors than PLST and CPLST, as the selected labels are in general easier to learn than the transformed labels. The encoding error, on the other hand, decreases with k as expected. In particular, PLST, which transforms $\mathbf{Y}$ to its best rank-k representation, yields the smallest encoding error.



(a) training error. (b) encoding error.

Figure 1. Variation of the training and encoding errors with k on cal500 .

5. Conclusion

In this paper, we proposed an efficient approach for handling multi-label classification problems with many labels. Using a label selection approach, we sample the more important labels by considering it as a column subset selection problem (CSSP). Instead of using a pre-determined number of sampling trials as in existing CSSP algorithms, the number of trials used in the proposed algorithm is adaptive. Empirically, a much smaller number of sampling trials is needed. Theoretical analysis shows that the proposed sampling approach is highly efficient. It can also obtain a good approximation of the label matrix, and a good multi-label classification performance bound. Experiments performed on a number of real-world data sets with large number of labels demonstrate that the proposed algorithm is effective and efficient as compared to the various recent multi-label learning algorithms.

Acknowledgments

This research was supported in part by the Research Grants Council of the Hong Kong Special Administrative Region (Grant 614012).

References

Bach, F., Jenatton, R., Mairal, J., and Obozinski, G. Convex optimization with sparsity-inducing norms. In Sra, S., Nowozin, S., and Wright, S. J. (eds.), *Optimization for Machine Learning*. MIT Press, 2011.

Balasubramanian, K. and Lebanon, G. The landmark selection method for multiple output prediction. In *Proceedings of the 29th International Conference on Machine Learning*, pp. 983–990, Edinburgh, Scotland, June 2012.

Bi, W. and Kwok, J.T. Multi-label classification on tree- and DAG-structured hierarchies. In *Proceedings of the 28th International Conference on Machine Learning*, pp. 17–24, Bellevue, WA, USA, June 2011.

Boutsidis, C., Mahoney, M.W., and Drineas, P. An improved approximation algorithm for the column subset selection problem. In *Proceedings of the 20th ACM-SIAM Symposium on Discrete Algorithms*, pp. 968–977, New York, NY, USA, January 2009.

Chen, Y-N. and Lin, H-T. Feature-aware label space dimension reduction for multi-label classification. In *Advances in Neural Information Processing Systems 25*, pp. 1538–1546, 2012.

Dembczynski, K., Cheng, W., and Hüllermeier, E. Bayes optimal multilabel classification via probabilistic classifier chains. In *Proceedings of the 27th International Conference on Machine Learning*, pp. 279–286, Haifa, Isreal, June 2010.

Drineas, P., Mahoney, M., and Muthukrishnan, S. Subspace sampling and relative-error matrix approximation: Column-based methods. In *Proceedings of the 9th International Conference on Approximation Algorithms for Combinatorial Optimization Problems*, pp. 316–326, Barcelona, Spain, August 2006.

Duygulu, P., Barnard, K., Freitas, N.D., and Forsyth, D. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In *Proceedings of the 7th European Conference on Computer Vision*, pp. 97–112, Copenhagen, Denmark, May 2002.

Gu, M. and Eisenstat, S.C. Efficient algorithms for computing a strong rank-revealing qr factorization. *SIAM Journal on Scientific Computing*, 17(4):848–869, 1996.

Hariharan, B., Zelnik-Manor, L., Vishwanathan, S.V.N., and Varma, M. Large scale max-margin multi-label classification with priors. In *Proceedings of the 27th International Conference on Machine Learning*, pp. 423–430, Haifa, Isreal, June 2010.

Hsu, D., Kakade, S.M., Langford, J., and Zhang, T. Multi-label prediction via compressed sensing. In *Advances in Neural Information Processing Systems 22*, pp. 772–780, 2009.

Mahoney, M.W. Randomized algorithms for matrices and data. *Foundations and Trends in Machine Learning*, 3 (2):123–224, 2011.

Mencía, E.L. and Fürnkranz, J. Efficient pairwise multilabel classification for large-scale problems in the legal domain. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 50–65, Antwerp, Belgium, 2008.

Rudelson, M. and Vershynin, R. Sampling from large matrices: An approach through geometric functional analysis. *Journal of the ACM*, 54(4):21, 2007.

Tai, F. and Lin, H.T. Multilabel classification with principal label space transformation. *Neural Computation*, 24 (9):2508–2542, 2012.

Tsochantaridis, I., Joachims, T., Hofmann, T., and Altun, Y. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, 6(2):1453, 2005.

Tsoumakas, G., Katakis, I., and Vlahavas, I. Effective and efficient multilabel classification in domains with large number of labels. In *Proceedings of the ECML/PKDD Workshop on Mining Multidimensional Data*, Antwerp, Belgium, 2008.

Tsoumakas, G., Katakis, I., and Vlahavas, I. Mining multilabel data. In Maimon, O. and Rokach, L. (eds.), *Data Mining and Knowledge Discovery Handbook*, pp. 667–685. Springer, 2010.

Turnbull, D., Barrington, L, Torres, D., and Lanckriet, G. Semantic annotation and retrieval of music and sound effects. *IEEE Transactions on Audio, Speech and Language Processing*, 16(2):467–476, 2008.

Vens, C., Struyf, J., Schietgat, L., Džeroski, S., and Blockeel, H. Decision trees for hierarchical multi-label classification. *Machine Learning*, 73:185–214, 2008.

Zhang, Y. and Schneider, J. Multi-label output codes using canonical correlation analysis. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*, pp. 873–882, Ft. Lauderdale, FL, USA, April 2011.

Zhang, Y. and Schneider, J. Maximum margin output coding. In *Proceedings of the 29th International Conference on Machine Learning*, pp. 1575–1582, Edinburgh, Scotland, June 2012.

Zhou, T., Tao, D., and Wu, X. Compressed labeling on distilled labelsets for multi-label learning. *Machine Learning*, 88:69–126, 2012.