Mach Learn (2012) 88:69–126
DOI 10.1007/s10994-011-5276-1

Compressed labeling on distilled labelsets for multi-label learning

Tianyi Zhou · Dacheng Tao · Xindong Wu

Received: 30 September 2010 / Accepted: 12 December 2011 / Published online: 6 January 2012

Abstract Directly applying single-label classification methods to the multi-label learning problems substantially limits both the performance and speed due to the imbalance, dependence and high dimensionality of the given label matrix. Existing methods either ignore these three problems or reduce one with the price of aggravating another. In this paper, we propose a $\{0, 1\}$ label matrix compression and recovery method termed "compressed labeling (CL)" to simultaneously solve or at least reduce these three problems. CL first compresses the original label matrix to improve balance and independence by preserving the signs of its Gaussian random projections. Afterward, we directly utilize popular binary classification methods (e.g., support vector machines) for each new label. A fast recovery algorithm is developed to recover the original labels from the predicted new labels. In the recovery algorithm, a "labelset distilling method" is designed to extract distilled labelsets (DLs), i.e., the frequently appeared label subsets from the original labels via recursive clustering and subtraction. Given a distilled and an original label vector, we discover that the signs of their random projections have an explicit joint distribution that can be quickly computed from a geometric inference. Based on this observation, the original label vector is exactly determined after performing a series of Kullback-Leibler divergence based hypothesis tests on the distribution about the new labels. CL significantly improves the balance

Editors: Grigorios Tsoumakas, Min-Ling Zhang, and Zhi-Hua Zhou.

T. Zhou · D. Tao
Centre for Quantum Computation & Intelligent Systems (QCIS), Faculty of Engineering & IT, University of Technology, Sydney (UTS), 235 Jones Street, Broadway, NSW 2007, Australia

T. Zhou
e-mail: tianyi.zhou@student.uts.edu.au

D. Tao
e-mail: dacheng.tao@uts.edu.au

X. Wu (✉)
Department of Computer Science, Hefei University of Technology, Hefei, 230009, China
e-mail: xwu@cs.uvm.edu

X. Wu
Department of Computer Science, University of Vermont, 351 Votey Hall, Burlington, VT 05405, USA

of the training samples and reduces the dependence between different labels. Moreover, it accelerates the learning process by training fewer binary classifiers for compressed labels, and makes use of label dependence via DLs based tests. Theoretically, we prove the recovery bounds of CL which verifies the effectiveness of CL for label compression and multi-label classification performance improvement brought by label correlations preserved in DLs. We show the effectiveness, efficiency and robustness of CL via 5 groups of experiments on 21 datasets from text classification, image annotation, scene classification, music categorization, genomics and web page classification.

Keywords Multi-label prediction · Labelset selection · Compressed sensing · Random projection · Label compression · Distilled labelsets · Binary matrix decomposition · KL divergence · Hypothesis test of distribution · Support vector machines

1 Introduction

The past years have witnessed significant contributions of multi-label learning for various practical applications, such as text classification, image annotation, scene classification, music categorization, genomics and web page classification, where each sample simultaneously belongs to several classes out of a great amount of possible candidates. Recently, learning from data with multiple labels attracts growing attention from related fields and is developed rapidly. Its importance and necessity have been well appreciated by plenty of specific utilizations.

In contrast to single-label binary classification, multi-label learning predicts a $\{0, 1\}$ label matrix $Y \in \{0, 1\}^{n\times k}$ (n is the number of samples and k is the number of labels) rather than a $\{0, 1\}$ label vector $y \in \{0, 1\}^n$. At an early stage, binary relevance (BR) (Tsoumakas and Katakis 2007) and label powerset (LP) (Tsoumakas and Katakis 2007) were developed to transform a multi-label learning problem to several binary classification tasks. Specifically, BR associates each label with an individual class, i.e., assigns samples with the same label to the same class; and LP treats each unique set of labels as a class, in which samples share the same label vector.

Although BP/LP and their variants can directly transform a multi-label learning problem into multiple binary classification tasks, it has been widely acknowledged that these transformations share the following three problems: sample imbalance, label dependence and label high-dimensionality. Because these transformations do not consider the differences between the information embedded in several independent single labels and multiple labels. These problems may ruin the binary classifier training and even end up with trivial solutions to the label prediction, e.g., assigning the same label to all samples from different classes.

1.1 Three problems

The problem of sample imbalance It usually occurs in multi-label learning and multi-class learning, when more than two classes are considered and the one-versus-all rule is adopted. In such a case, the conventional binary classification methods tend to overwhelm the class with more samples. This sample imbalance between these two classes will seriously weaken the classification performance and even make the learning task fail by assigning the same label to all samples from different classes.

Specifically, BR directly uses the original label matrix as the class indicator matrix and trains a classifier for each label. It allows the overlapping of classes and treats them as independent ones. Thus BR has the imbalance problem when 0 and 1 in columns of the

label matrix are imbalanced. LP treats each unique labelset as the sign of an independent class and transfers a multi-label prediction problem to a larger size multi-class classification problem. In contrast to BR, LP tremendously increases the number of classes and decreases samples in each class, so it aggravates the sample imbalance problem.

The problem of label dependence This is a characteristic problem of multi-label learning and tells the difference between multi-label learning and multi-class learning, because it is admitted a simultaneous appearance of different labels on one sample. The dependence or correlation between different labels can then be studied by using statistics of their distributions, e.g., χ^2 test and Pearson's correlation coefficient. This dependence between labels ends up with the poor performance of multi-label learning when it is directly decomposed into several binary classification tasks. That is because the direct decomposition assumes different labels are independent of each other and ignores the label dependence. In contrast to binary classification, samples sharing one identical label in multi-label learning may quite differ in concepts and have large pairwise distances in the feature space, because their other labels can be different. There are two types of label dependence (Dembczyński et al. 2010), the conditional dependence and unconditional dependence. The former captures the label dependence given a specific sample, while the latter considers the global label dependence in the label space. Exploiting label dependence becomes a popular motivation in recently developed multi-label learning methods (Read et al. 2009; Bianchi et al. 2006; Tsoumakas et al. 2008). Both empirical and theoretical studies have proved that it helps improving the learning performance by considering the label dependence.

BR simply ignores the dependence between labels and independently trains classifiers for given labels. Thus it performs unsatisfying when the labels are highly correlated to each other. LP treats the unique different labelsets as independent classes and trains one binary classifier for each of them. It takes label dependence into consideration, but it neglects the correlation or shared labels between different labelsets and deteriorates the data imbalance as the cost.

The problem of label high dimensionality In practical problems, e.g., text classification (Gomez et al. 2011; Liu and Liang 2011) and image annotation, data usually have hundreds or even thousands of labels, which leads to a $\{0, 1\}$ label matrix with sparse entries and high dimensionality. The high dimensionality of the label matrix makes the multi-label learning task very challenging. In particular, the increasing of label dimensionality enlarges the sample imbalance in each binary classification and increases the number of classifiers to be trained.

BR directly adopts the original label matrix, while LP increases the number of labels to K much larger than that of the original labels. Most existing multi-label learning methods transfer the original problem to m binary classifications, wherein m is a number between k and K.

1.2 Previous works

Both empirical and theoretical studies of the existing multi-label algorithms suggest that the learning performance is determined by specific properties of a multi-label dataset, e.g., label dependence, label structure and dependence between samples and the corresponding labels. Nevertheless, most existing methods either partially tackle some of the three problems and ignore the others, or eliminate one with the price of exacerbating the other two. In this paper, we categorize these methods into five groups. It is impossible to exhaustively summarize all

the published methods. Thus, domain experts will easily note the missing references. We hope that the cited reviews (Tsoumakas et al. 2010; Tsoumakas and Katakis 2007) cited here will point to the missing references.

Label transformation Methods belonging to this group transform the given labels into new ones, and then decompose the original multi-label prediction problem into a series of binary classification tasks according to the new labels. This group of methods can embed the label dependence information into the transformed labels, and can exploit the label structure to decrease the number of new labels.

Some methods, e.g., BR and LP, in this group treat new labels independently, so the label transformation and the classifier training are independent. After obtaining the new labels, one binary classifier is trained for each new label independently without considering its relationship to others. The pruned problem transformation (PPT) (Read et al. 2008) modifies LP via replacing the rare labelsets with their more frequent subsets, and thus both the sample imbalance problem and label high dimensionality problem are alleviated. The random k-labelsets (RAkEL) method (Tsoumakas and Vlahavas 2007) randomly selects an ensemble of subset from the original labelsets, and then LP is applied to each subset. The sequential prediction is accomplished by ranking and thresholding on the results of the ensemble of LP classifiers. It is a modification of LP with the motivation of utilizing the label dependence. Ranking by pairwise comparison (RPC) (Hüllermeier et al. 2008) adopts one-versus-one rule by training a binary classifier for each pair of labels and ranking the classification results for prediction. RPC alleviates the sample imbalance problem in the training of each classifier, but increases the number of labels to $k(k-1)/2$ (k is the number of original labels). Related methods includes multi-label pairwise perceptron (MLPP) (Mencía and Fürnkranz 2008) and calibrated label ranking (CLR) (Fürnkranz and Hüllermeier 2008).

Other methods in this group establish a structure of labels. The classifier training and label prediction are then implemented on the obtained structure. These methods take the label dependence into consideration and reduces the sample imbalance problem. Two representatives are hierarchical binary relevance (HBR) (Bianchi et al. 2006) and hierarchy of multi-label classifiers (HOMER) (Tsoumakas et al. 2008). HBR builds a general-to-specific tree structure of labels, where a sample with a label must be associated with its parent labels. For each non-root label on the hierarchical structure, a binary classifier is designed by using a subset of samples whose labels include the parent labels of the current one. HOMER recursively partitions the labels into several subsets and builds a tree hierarchy. In the hierarchy, each node is composed of several labels that are separated into a number of subsets in its child nodes. HBR method is then applied to each node for obtaining multi-label classifiers to separate the child nodes.

Regularized classifications This group of methods formulates the problem as a series of classifications with regularization. Stacking method (Cheng and Hüllermeier 2009) and "Curds and Whey (C&W)" procedure (Breiman and Friedman 1997) separate the classification and regularization as two isolated stages. They train a classifier on each label as BR, and then correct the prediction of each label by using the predictions of the others. These two methods impose a regularization to the conventional classification results, wherein the predictions of the other labels perform as a bias to decrease the variance of current label prediction. The regularization item always aims to exploit label dependence. Another kind of regularization directly solve regularized classification problems and jointly learn all the binary classifiers that share a parameter space. Two examples are regularized multi-task learning (Evgeniou and Pontil 2004) and shared-subspace learning (Jia et al. 2010). If a linear classifier w_i is trained on each label, the former method assumes $w_i = v_i + w_0$, while

the latter method assumes $w_i = v_i + u_i \Theta$. Both the w_0 and Θ are shared parameters that store the label dependence information.

Multi-label learning problem reformulation This group of methods formulates the multi-label learning problem as other supervised learning problems (Kong and Yu 2011) rather than classification and ranking, which are two methods usually extended from single-label learning. Multi-label dimensionality reduction via dependence maximization (MDDM) (Zhang and Zhou 2008) tackles the "curse of dimensionality" in multi-label data and formulates the problem as a discriminative dimension reduction (Zhou et al. 2011; Bian and Tao 2011). It maximizes the dependence between feature space and label space via maximizing the empirical estimate of Hilbert-Schmidt Independence Criterion (HSIC) (Gretton et al. 2005). Graphical models such as conditional random fields (CRF) (Ghamrawi and McCallum 2005) are natural solutions to estimate the joint distribution of samples and labels in multi-label learning. They provide a probabilistic formulation of the problem. The classifier chain (CC) (Read et al. 2009) adopts a greedy way to concatenate the binary classifiers for all the labels and makes use of conditional label dependence. It trains a classifier for each label at a time by using given samples and the previously predicted labels as the input. Thus the prediction of each label is related to the previously predicted ones. It has an ensemble variant (ECC) (Read et al. 2009) and a probabilistic variant (PCC) (Dembczyński et al. 2010). The former alleviates the influence of label order, while the latter tackles the Bayes-optimal solution of CC.

Linear regression This group of methods adopts linear regression model to solve multi-label learning problems. Although linear model for classification is criticized due to its underlying assumption (Hastie et al. 2009), a number of popular techniques can be used to solve the aforementioned three problems in this scenario. In particular, the linear model is $Y = XW$, where Y is the label matrix and the columns of W are the corresponding model coefficient vectors. In Jia and Ye (2009), the optimization of W is formulated as a matrix completion problem when W is assumed to be low-rank. The given samples and the corresponding label vectors comprise the random measurement matrix ensemble $X_i^T Y_i$ $(i = 1, \ldots, n)$. The low rank assumption of W embeds the label dependence in the learning process. A multi-task method proposed in Chen et al. (2010) assumes that W is the sum of a low-rank component and a sparse component. Different from imposing an extra assumption on W in the linear model, another observation is that the $\{0, 1\}$ label matrix Y is sparse and thus compressible. In Hsu et al. (2009), Y is compressed via random projections $Y' = YA$, and then a new regression model $Y' = XW'$ is obtained, the original Y can be recovered via compressed sensing algorithms. This is a successful application of compressed sensing (CS) (Donoho 2006) to multi-label learning and inherits the theoretical merit of CS, i.e., only $\mathcal{O}(\log(k))$ models needs to be trained for data with k labels. This method reduces the model complexity caused by the large number of labels.

Algorithm extension This group of methods extends or modifies the existing supervised learning algorithms to the multi-label learning scenario. C4.5 is a popular decision tree algorithm and is extended to multi-label learning in Clare and King (2001) via modifying the entropy criterion. AdaBoost has been extended to multi-label data ranking and Hamming loss minimization in AdaBoost.MR and AdaBoost.MH (Schapire and Singer 2000), respectively. The multi-class multi-label perceptron (MMP) (Crammer and Singer 2003), back-propagation for multi-label Learning (BP-MLL) (Zhang and Zhou 2006), and RBF neural networks for multi-instance multi-label learning (MIMLRBF) (Zhang

and Wang 2009) are extensions of neural networks algorithms in multi-label learning. Multi-label k-nearest neighbor (ML-knn) (Zhang and Zhou 2007) is an extension of knn. It obtains the label prior distribution from the k nearest neighbors and applies "maximizes a posteriori (MAP)" to the label prediction. It partially solves the imbalance problem.

Multi-label learning methods can also be roughly distinguished into learning reduction methods and fully-specified learning methods. In particular, the "label transformation" and "algorithm extension" in our results can be attributed to "learning reduction methods" because they transform the multi-label learning problem into other different subproblems. "Regularized classification", "multi-label learning problem reformulation", "linear regression" and some methods of "algorithm extension" in our results can be attributed to "specified learning methods" because they formulate multi-label learning as specific problems. From some perspective, compressed labeling (CL) proposed in this paper can be viewed as a learning reduction method.

1.3 The proposed method

In this paper, we propose a $\{0, 1\}$ label matrix compression and recovery scheme termed "compressed labeling (CL)" for multi-label learning. It simultaneously solves or at least substantially alleviates the aforementioned three problems via random coding of the label matrix and fast corresponding decoding (recovery). CL is a general scheme for embedding existing single-label learning methods in a multi-label learning setting. The label compression in CL leads to a shrinkage of the problem size, and thus it is efficient in large-scale problems (Masud et al. 2011).

We summarize the CL scheme including its training stage and prediction stage in Fig. 1.

In the training stage, CL first compresses the given $\{0, 1\}$ label matrix Y into a sign matrix Z of its random projections on Gaussian random matrix A. Due to the properties of random projections, the new labels in Z are independent of each other and the sample imbalance problem for each class is substantially alleviated. Afterward, one binary classifier, e.g., SVM, is trained for each new label independently on the training set $\{X, Z\}$.

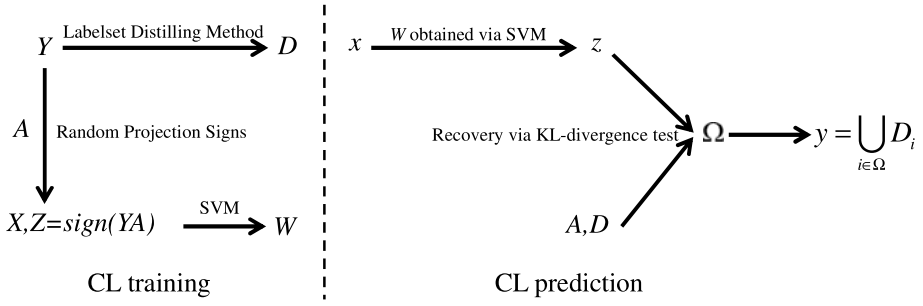

Fig. 1 Compressed labeling on distilled labelsets. In the training stage, CL first compresses the original label matrix Y into Z, which is the sign matrix of random projections of Y on Gaussian random matrix A. Then binary classifiers (such as SVM) corresponding to the training set $\{X, Z\}$ are independently learned and stored in W. Meanwhile, the frequently appeared label subsets in Y are extracted by labelset distilling method (LDM) and stored in the distilled labelsets (DLs) D. In the prediction stage, CL first predicts the new labels z of a given sample x via the binary classifiers W. Given A and D, the DLs appearing in z are identified by a KL-divergence test based recovery algorithm and indexed by Ω. The final prediction y is the union of all the appeared DLs

In the prediction stage, CL first predicts the new labels z of a given sample x via the binary classifiers W obtained in the training stage, and then a fast recovery algorithm is developed to recover the original labels y from the predicted new labels z. In the recovery algorithm, we predict the original label vector of the given sample via performing a series of KL divergence based hypothesis tests on the "distilled labelsets (DLs)".

DLs are the frequently appeared label subsets extracted from the original labels Y via a "label distilling method (LDM)". LDM performs a recursive clustering and subtraction on the label vectors, i.e., the rows of Y. Each distilled labelset (DL) is the intersection of the label vectors in each cluster. LDM takes the label dependence into consideration and this critical information guarantees the success of the recovery algorithm.

Given a DL and an original label vector, we discover that the signs of their random projections on a Gaussian ensemble follow an explicit joint distribution that can be quickly computed from a geometric inference. The corresponding empirical joint distribution can also be quickly obtained from the new label matrix. A KL divergence based comparison between the explicit joint distribution and the empirical one indicates whether a given DL is a subset of the original label vector. Since this test includes only comparison and thresholding, the recovery algorithm is fast with linear time complexity.

We theoretically prove the recovery bound of CL by investigating the upper bounds for the probabilities of 2 types of recovery failures in CL. The probabilistic bound for recovery failures exponentially shrinks with the increasing of measurements, i.e., the dimensionality of the compressed label matrix, and with the increasing of the cardinalities of the distilled labelsets. This result soundly shows the effectiveness of label compression and the prediction improvement brought by DL.

We evaluate CL on both large-scale datasets and small-scale ones including text classification, music categorization, image annotation, scene classification, genomics and web data mining. The experiments are divided into 5 groups on 21 datasets. The first group tests the label matrix compression and recovery. The recovery accuracy, sample balance and mutual independence of new labels are evaluated. The other 4 groups test CL in multi-label prediction problems, and five different evaluation metrics are used to measure its prediction performance. Thorough comparisons between CL and BR, 3 popular multi-label learning methods, 2 SVM algorithms dealing with imbalance datasets are provided respectively. The trade-off between label compression and prediction improvement is empirically studied and analyzed. The experimental results show the effectiveness, efficiency and robustness of CL in multi-label learning.

The rest of the paper is organized as follows. Section 2 presents the label matrix compression and classification in CL. Section 3 presents LDM and the KL divergence test based recovery algorithm of CL. Section 4 studies the relationship between compressed labeling and compressed sensing, and the contribution of CL to multi-label learning. Section 5 shows the experimental results. Section 6 concludes the paper.

2 Compressed labeling via random projections

This section presents the label matrix compression and the subsequent classification in CL, which comprises the training stage of CL. The label matrix compression is based on the random projections of the given $\{0, 1\}$ label matrix on a Gaussian ensemble. We show that the pseudorandomness of the new label matrix results in (1) an improved balance of samples for the binary classification on each label, (2) the mutual independence among the new labels and (3) the information of original label matrix is properly preserved in a low-dimensional

subspace. These improvements solve or at least reduce the three aforementioned problems in Sect. 1.1 and thus benefit simultaneously the subsequent classification.

2.1 Random projection signs of label matrix

Random projection (Vempala 2004) is a simple yet powerful technique that has been widely used in fast approximation algorithms and data recovery. It is introduced as an efficient pairwise distance calculation and approximation method according to Johnson-Lindenstrauss (JL) Lemma (Johnson and Lindenstrauss 1984) and its variants in particular metric spaces, i.e., ℓ_2 (Euclidian) space (Vempala 2004), ℓ_p $(0 < p < 2)$ space (stable random projection) (Li 2008), $\ell_p (p > 2)$ (high-order) space (Li 2010) and smooth manifold (Clarkson 2008). This property has been broadly used in fast nearest neighbor search (Indyk and Motwani 1998), low distortion embedding (Dasgupta 2000) and hashing (Dasgupta and Freund 2008). Compressed sensing (Candès et al. 2006) proves that an exact reconstruction of a sparse signal from a few of its random projections is possible, when the random projection ensemble satisfies restricted isometry property (RIP). Random projection also attracts attentions in matrix low-rank approximation, because the column space of a matrix's low-rank random projection is proved as a sufficiently close approximation of its principle subspace (Halko et al. 2009).

In CL, the random projection offers a different function. Since CL formulates the multi-label prediction as a classification problem rather than a linear regression problem, the label matrix after compression has to be a binary matrix rather than a real-valued one. Thus the direct utilization of random projection is improper. We consider the signs of the random projections in CL, so the compressed label matrix Z is:

$$Z = \operatorname{sign}(YA), \tag{1}$$

where $Z \in \{-1, 1\}^{n\times m}$ is the compressed label matrix, $Y \in \{0, 1\}^{n\times k}$ is the original label matrix, $A \in \mathbb{R}^{k\times m}$ is a random matrix whose columns are randomly sampled from an ensemble, and $\operatorname{sign}(\cdot)$ is an element-wise sign operator. In CL, we adopt the i.i.d. standard Gaussian ensemble, i.e., entries of A are independent standard normal variables.

Although it seems that the hard thresholding of random projections in (1) discards partial useful information for recovery at the first glance, the information is fully retained in DLs and the $\{0, 1\}$ binary prior of the label matrix, which play critical roles in the CL recovery algorithm.

This simple label compression method provides an effective cure for the aforementioned three problems, i.e., sample imbalance, label dependence and label high dimensionality. It is self-evident that the last problem is alleviated on the CL labels, because the number of random projections m can be much less than k in CL. Therefore, the number of binary classifiers in the training stage can be substantially reduced from k to m, which significantly reduces the computational complexity.

Below, we theoretically show that CL improves the sample balance and ensures the label independence. Empirical studies of these two properties on several datasets are presented in the experimental section.

2.2 Improved sample balance of CL labels

Given a set of positive samples labeled by 1 and negative ones labeled by 0 for each label in multi-label learning, without loss of generality, we can define the degree-of-balance of

a given label matrix $Y \in \{0, 1\}^{n\times k}$ as the average proportion of positive samples on all the different labels, i.e.,

$$balance = \frac{1}{k}\sum_{i=1}^{k}\frac{np_i}{n}, \tag{2}$$

where np_i is the number of positive samples on the ith label. The degree-of-balance close to 0.5 yields a balanced sample set for training.

In CL labels, the degree-of-balance on each label has essential connection with the overlaps between unique labelsets of the original labels. Before investigating this phenomenon, we first give an important theorem that will be used several times in this paper. The main significance of this theorem is that it bridges the inner distribution of the compressed label vector z (an arbitrary row of Z) with the corresponding original label vector x via a one-to-one bijection. This bijection provides a direct and efficient estimate to x from the statistics of z, which will be presented in Sects. 3.2 and 3.3.

Theorem 1 (Random projection signs of two binary vectors) *Given two nonzero binary vectors $x, y \in \{0, 1\}^k$, the signs of their projections on a random vector α satisfies the following distribution, if the entries of α are independent standard Gaussian variable with unit variance and zero mean.*

$$\Pr(\text{sign}(\langle x, \alpha\rangle) \cdot \text{sign}(\langle y, \alpha\rangle) = -1) = \frac{1}{\pi}\arccos\left(\frac{\text{card}(x \cap y)}{\sqrt{\text{card}(x)}\sqrt{\text{card}(y)}}\right), \tag{3}$$

where $\text{card}(\cdot)$ *refers to the cardinality of a given vector.*

Proof The proof follows a geometric inference on a sphere in a high dimensional space. Since the entries of α are i.i.d. standard Gaussian variables, α is a vector which is drawn uniformly from a hypersphere S_k in the k dimensional space. Figure 2 shows the random projections of x, y on two Gaussian random vectors α and β.

In Fig. 2, we uses "+" and "−" to indicate the signs of random projections. Two hyperplanes $W1$ and $W2$ are perpendicular to x and y, respectively. Figure 2 verifies that if x and y are projected onto a random vector β in the two shaded regions, which determined by $W1$ and $W2$, the random projections will have opposite signs. When x and y are projected onto a random vector α in the unshaded regions, their random projection signs will be identical.

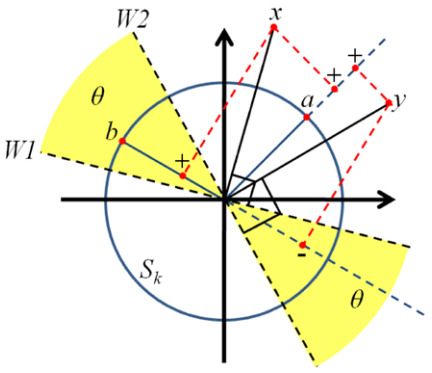

Fig. 2 Random projections of x, y on two random vectors α and β, which are drawn uniformly from a k-dimensional hypersphere. The signs of random projections are marked as "+" for positive and "−" for negative in the figure. The hyperplanes $W1$ and $W2$ are perpendicular to x and y, respectively

Since the dihedral angle θ is equal to the angle between vectors x and y, i.e.,

$$\theta = \arccos\left(\frac{\langle x, y\rangle}{\|x\|_2\|y\|_2}\right) = \arccos\left(\frac{\text{card}\,(x \cap y)}{\sqrt{\text{card}(x)}\sqrt{\text{card}(y)}}\right), \tag{4}$$

and the Gaussian random vector α is drawn uniformly from the hypersphere, the probability that x and y have different random projection signs is proportional to the area of the shaded regions. Therefore, we have

$$\Pr(\text{sign}\,(\langle x, \alpha\rangle) \cdot \text{sign}\,(\langle y, \alpha\rangle) = -1) = \frac{2\theta}{2\pi}, \tag{5}$$

which completes the proof of Theorem 1. □

Theorem 1 can be seem as an extension of Lemma 3.2 in Goemans and Williamson (1995). Now we analyze the degree-of-balance of CL labels based on Theorem 1. Without loss of generality, we assume the original label matrix Y has the following label powerset L, which consists of the unique rows of Y:

$$L = [L_1; L_2; \ldots; L_K], \quad L_i \in \{0, 1\}^k. \tag{6}$$

We call each unique row of Y as a "labelset". The appearance times of labelset L_i in all the rows of Y is represented as n_i, so we have $n = \sum_{i=1}^{K} n_i$. Given a vector α whose entries are randomly drawn from independent standard Gaussian distributions, if the random projection sign of L_i on α, i.e., $z_{l:Y_l=L_i} = \text{sign}(\langle L_i, \alpha\rangle)$ in the corresponding column of Z is known, the probabilities that random projection signs of the other label vectors $L_{j:j\neq i}$ in L are $-z_l$ can be obtained by using (3). In particular, we calculate the expected number of label vectors (rows) in Y whose random projection signs on an arbitrary row α of A are opposite to z_l, namely, the expected number of $-z_l$ in arbitrary column z of CL label matrix Z:

$$\begin{aligned}
\mathbb{E}_{z_l}\left(\left|p : z_p = -z_l\right|\right) &= \sum_{j=1, j\neq i}^{K} n_j \Pr\left(z_l \cdot \text{sign}\left(\langle L_j, \alpha\rangle\right) = -1\right) \\
&= \sum_{j=1, j\neq i}^{K} n_j \Pr\left(\text{sign}\,(\langle L_i, \alpha\rangle) \cdot \text{sign}\left(\langle L_j, \alpha\rangle\right) = -1\right) \\
&= \sum_{j=1, j\neq i}^{K} \frac{n_j}{\pi} \arccos\left(\sqrt{\frac{\text{card}(L_i \cap L_j)}{\text{card}(L_i)}} \cdot \sqrt{\frac{\text{card}(L_i \cap L_j)}{\text{card}(L_j)}}\right),
\end{aligned} \tag{7}$$

where $|\cdot|$ denotes the number of given variable, and the last step is due to Theorem 1.

The conditional expectation in (7) computes the expected number of samples with CL label $-z_l$ in an arbitrary column of Z given the CL label of L_i, i.e., z_l. Without loss of generality, we assume $z_l = -1$. Thus the expected degree-of-balance $\mathbb{E}_{z_l=-1}(balance)$ given $z_l = -1$ can be calculated by using (7). Since the distribution of L_i in rows of Y is given by

$$\Pr(L_i) = n_i/n, \tag{8}$$

the unconditioned expected degree-of-balance $\mathbb{E}(balance)$ can be computed over the whole label powerset L, i.e.,

$$\begin{aligned}\mathbb{E}(balance) &= \sum_{i=1}^{K} \Pr(L_i)\mathbb{E}_{z_l=-1}(balance) = \sum_{i=1}^{K} \Pr(L_i) \cdot \frac{np_i}{n} \\ &= \sum_{i=1}^{K} \frac{n_i}{n} \cdot \frac{1}{n}\mathbb{E}_{z_l}\left(\left|p : z_p = -z_l\right|\right) \\ &= \sum_{i=1}^{K} \sum_{j=1, j\neq i}^{K} \frac{n_i n_j}{n^2 \pi} \arccos\left(\sqrt{\frac{\text{card}(L_i \cap L_j)}{\text{card}(L_i)}} \cdot \sqrt{\frac{\text{card}(L_i \cap L_j)}{\text{card}(L_j)}}\right). \end{aligned} \tag{9}$$

In multi-label learning, a degree-of-balance close to 0.5 is preferred, because the numbers of positive samples and negative ones will be equal to each other in the training set. To see how CL improves the sample balance, we first study a special case of multi-label learning: multi-class learning. In multi-class learning, each labelset in L includes only one "1" in its entries. There is no overlap between any two different labelsets in multi-class learning, which is a special case of multi-label learning. When the number of labelsets K in L increases, the label matrix Y seriously confronts the problem of sample imbalance. That is because each class has only a few positive samples and a large amount of negative ones. In such case, we have:

$$\sum_{i=1}^{K} \sum_{j=1, j\neq i}^{K} \frac{n_i n_j}{n^2} \to 1, \tag{10}$$

$$\sqrt{\frac{\text{card}(L_i \cap L_j)}{\text{card}(L_i)}} = 0, \quad (i \neq j). \tag{11}$$

Equation (10) is a result of a large K, while (11) is due to the orthogonality of the unique labelsets. Thus the expected degree-of-balance of CL labels in multi-class learning problem can be calculated by substituting the above equations into (9):

$$\mathbb{E}_{multi\text{-}class}(balance) = \frac{1}{2} \sum_{i=1}^{K} \sum_{j=1, j\neq i}^{K} \frac{n_i n_j}{n^2} \to 0.5. \tag{12}$$

Therefore, after label compression in CL, the problem of sample imbalance in multi-class learning is substantially alleviated.

The situation in general multi-label learning is similar but different because of the existence of labelset overlapping. Without loss of generality, we consider a pair of labelsets L_i and L_j. Referring to the situation of multi-class learning, small values of

$$\sqrt{\frac{\text{card}(L_i \cap L_j)}{\text{card}(L_i)}} \quad \text{and} \quad \sqrt{\frac{\text{card}(L_i \cap L_j)}{\text{card}(L_j)}} \tag{13}$$

are preferred in multi-label learning, because the arccosine of their multiplication will be close to $\pi/2$. Thus the expected degree-of-balance in (9) will approach to 0.5. This corresponds to a small overlap $\text{card}(L_i \cap L_j)$ or large cardinalities $\text{card}(L_i)$ and $\text{card}(L_j)$, which

are exactly in accordance with the multi-label data. That is because in multi-label data, a labelset L_i with large overlap to the other ones usually has very large cardinality, while a labelset L_i with small cardinality often has ultra-small overlap to the other ones. In both of these two cases, the two values in (13) will be kept close to 0.

Even when L_i and L_j share a large overlap and both have small cardinality, the degree-of-balance of CL labels will not deviate far away from 0.5. That is because this case causes small n_i and n_j, which results in a small weight $n_i n_j / n^2 \pi$ in (9) to eliminate the influence of small arccosine in (9). Therefore, $\mathbb{E}(balance)$ approaches to 0.5.

We place an empirical study of the sample balance of CL labels on various multi-label datasets in the experimental section. The result demonstrates that the sample degree-of-balance after label compression in CL is significantly improved and is close to the ideal value 0.5.

2.3 Mutual independence of CL labels

The mutual independence between CL labels after label compression is a natural result of random projection sign. To see this, we have the following theorem:

Theorem 2 (Label independence) *Given a binary and nonzero vector $x \in \{0, 1\}^k$, its CL labels obtained via random projection signs $z = \text{sign}(xA)$ are independent random variables, if A is a standard Gaussian matrix with entries drawn from independent standard Gaussian distribution.*

Proof Consider two random variables $y_i = xA_i$ and $y_j = xA_j$, if A_i and A_j are both composed by independent standard Gaussian variables, y_i and y_j are weighted sum of independent standard Gaussian variables. Thus the two variables y_i and y_j are independent Gaussian variables. Since $z_i = \text{sign}(y_i)$ and $z_j = \text{sign}(y_j)$, the random variables z_i and z_j are independent. Therefore, the CL labels in z are mutual independent random variables. Two similar analyses can be found in Halko et al. (2009), Vempala (2004). This completes the proof. □

The mutual independence between CL labels after compression is not equal to a sheer discard of the label dependence information in the original labels. Actually, we extract and store the label dependence information in the distilled labelsets (DLs) that consists of the most frequent label subsets in the original label matrix, and use it in the recovery algorithm. We transform the original labels into independent ones before the training stage, because the conventional binary classification methods cannot use the label dependence, and will be even harmed by the label dependence under some circumstances. Therefore, CL isolates the application of label dependence from the training stage in its scheme in order to apply binary classification methods without loss of useful information.

The label compression preserves the distribution and the pairwise distance of the original label vectors in the low-dimensional space, though it reduces the number of labels. The pairwise-distance preservation of random projections has been proved in various scenarios (Vempala 2004). Its variant, the random projection signs, has also been proved as a pairwise-distance preservation method in terms of the cosine distance (Raginsky and Lazebnik 2009; Goemans and Williamson 1995). Therefore, the classifications on the compressed labels will not be more difficult than on the original labels.

2.4 Classification via support vector machines

After obtaining the CL label matrix Z via random projections of the original label matrix Y, we train one binary classifier for each new label on the CL label matrix Z. There are a large number of binary classification methods with appealing performance and fast speed. Most of them can be directly applied to the training stage of CL.

Among the existing binary classification methods, support vector machine (SVM) (Vapnik 1995) has been widely used because of its appealing properties in statistical machine learning theory, optimality in classification hyperplane design, robustness to different types of data, extensionality to nonlinear kernel space, and many existing fast solvers. In this paper, we adopt SVM as the binary classification method used in the training stage of CL.

3 Recovery algorithm on distilled labelsets

In this section, we introduce the label distilling method (LDM) and the label recovery algorithm in CL, which comprise the prediction stage of CL. In recovery, given the CL label vector z predicted by the m binary classifiers from a sample x and a $\{0, 1\}$ binary dictionary D for label vectors, CL predicts the original label vector y by testing whether each binary vector D_i is included in y, i.e., whether $D_i \cap y = D_i$ and then recovers $y = \bigcup_{i \in \Omega} D_i$ (Ω is the index set of D_i included in y). The recovery is based on the fact that the random projection signs of y and D_i have explicit joint distributions in two cases, i.e., $D_i \cap y = D_i$ and $D_i \cap y = \emptyset$. In the recovery algorithm, a hypothesis test is designed to decide whether $D_i \cap y = D_i$ by comparing the Kullback-Leibler (KL) divergence (Kullback and Leibler 1951) between the empirical joint distribution of the random projection signs and the two explicit joint distributions in the two cases. An available and natural choice of D_i is unit vector. In this case, the CL recovery algorithm element-wisely recovers y, i.e., each label is recovered independently. However, in CL, we develop LDM to obtain D. In particular, LDM extracts the most frequent label subsets, i.e., distilled labelsets (DLs) $D \in \{0, 1\}^{d \times k}$, by recursive clustering and subtraction of the label vectors (rows) in the label matrix Y of training set. Therefore, CL exploits label dependence via jointly recovering the correlated labels in y on DLs. We will show that DLs improves the accuracy of the recovery algorithm by increasing the KL divergence between the two explicit joint distributions in the two cases.

3.1 Labelset distilling method (LDM)

The discrete patterns frequently appearing in the label matrix Y refers to the label subsets that are frequently shared by the rows of Y. These patterns reveal the structural information of the binary matrix Y and label dependence embedded in the given label vectors.

In multi-label learning, BR simply ignores the label dependence information, while LP treats the unique labelsets independently and thus ignores the shared information of different label vectors. Recently, several methods have been developed to exploit the label dependence by building a tree-structural hierarchy for the labels. However, the correlation (e.g., co-occurrence and mutual exclusion) between two labels are probabilistic rather than deterministic. Therefore, it is hard to prune the tree hierarchy without discarding the minority instances. A tree hierarchy that retains most leafs explains few label dependence information and will significantly increase the problem size.

We propose "labelset distilling method (LDM)" method to exploit the correlations between the unique labelsets of L rather than single labels. LDM can be interpreted as a greedy

search for the frequent discrete patterns of L. It decomposes $\{0, 1\}$ label matrix L as:

$$L = UD, \quad U \in \{0, 1\}^{K \times d}, D \in \{0, 1\}^{d \times k}. \tag{14}$$

The obtained dictionary D is called "distilled labelsets (DLs)", each row of which is a "distilled labelset (DL)".

In LDM, the above decomposition is accomplished by a greedy search of discrete patterns. The greedy search is a recursive clustering and subtraction of the labelsets (rows) in L. It is described by the following procedure:

1. The rows of L are clustered by using an existing clustering algorithms, e.g., spectral clustering (Luxburg 2007; Ng et al. 2001) or k-means (MacQueen 1967). In our experiments, we use spectral clustering.
2. The shared binary pattern of each cluster is extracted as a row D_i in the DLs D and then subtracted from the labelsets L_i in the cluster.
3. For the clusters without shared pattern, labelsets in them are kept the same in the label powerset L. For the labelsets L_i that become all-zero vectors after substraction, we remove them from L. The other labelsets L_i after subtraction are updated in L.
4. Update coefficients in U corresponding to the newly extracted atoms in D.

The above procedure is iterated until the label powerset L is empty, i.e., all the unique labelsets L_i in the initial L are completely represented by the atoms in the dictionary D.

We use spectral clustering to group the rows of L, because the number of clusters obtained by spectral clustering in each iteration can be adaptively adjusted by a given threshold τ. In particular, we sort q eigenvalues of the Graph Laplacian from small to large, and compute the following metric for each one:

$$e_i = \frac{\sum_{j=i}^{q} \lambda_j}{\sum_{j=1}^{q} \lambda_j}, \quad i = 2, \ldots, q. \tag{15}$$

Only the eigenvectors with $e_i < \tau$ are selected for the subsequent processing (including k-means and thresholding). The number of selected eigenvectors is the number of clusters in the iteration. A properly selected parameter τ can efficiently generate clusters with shared labelsets. Empirically, we select $0.01 \leq \tau \leq 0.25$ in all experiments. This fact will be verified in our experiments.

We show LDM in Algorithm 1.

3.2 Joint distribution of two random projection signs

An interesting phenomenon in CL is that given a label vector L_i, the signs of its projection and an arbitrary D_i's projection on a standard Gaussian random vector α have an explicit joint distribution that can be quickly computed. Based on this fact, the existence of a distilled labelset D_i in an unknown label vector y can be tested by using the information of their random projection signs.

In particular, the following theorem states the random projection signs of y and D_i follows an explicit joint distribution.

Theorem 3 (Joint distribution of random projection signs) *Given a label vector $y \in \{L_1, L_2, \ldots, L_K\}$ that is selected from the rows of the label powerset L, let D be L's distilled labelsets (DLs) that satisfy $L = UD$. If D_i is included in y, i.e., $D_i \cap y = D_i$, the signs*

Algorithm 1: Labelset distilling method (LDM)

Input: Label powerset $L \in \{0,1\}^{K\times k}$, threshold τ
Output: Distilled labelsets D, coefficient matrix U
Initialize: $D := \emptyset$, $U := \emptyset$
while *The rows of L are not empty* **do**
  Cluster the rows of L into t clusters by `Spectral Clustering` with threshold τ;
  for $i \leftarrow 1$ **to** t **do**
    Extract the label subset $D_i \in \{0,1\}^k$ shared by the label vectors in *Cluster* i, i.e., $D_i = \bigcap_j \{L_j : L_j \in \textit{Cluster } i\}$;
    Subtract the obtained D_i from all the label vectors in *Cluster* i, i.e., $L_j := L_j - D_i$ for $\{L_j : L_j \in \textit{Cluster } i\}$;
    if $\{L_j : L_j \in \textit{Cluster } i\} = \boldsymbol{0}$ **then**
      Remove L_j from L
    end
    Add the extracted D_i into the distilled labelsets D as a new row;
    Add the corresponding coefficient vector $U_i^T \in \{0,1\}^K = \{U_{ij}^T = 1 \text{ if } L_j \in \textit{Cluster}, \text{else } U_{ij}^T = 0\}$ into the coefficient matrix U as a new column;
  end
end

of their random projections on a standard Gaussian random vector α *follows the following joint distribution* $P1$:

$$P1(1) = \Pr(\operatorname{sign}(\langle y, \alpha\rangle) \cdot \operatorname{sign}(\langle D_i, \alpha\rangle) = -1) = \frac{1}{\pi}\arccos\left(\sqrt{\frac{\operatorname{card}(D_i)}{\operatorname{card}(y)}}\right), \tag{16}$$

$$P1(2) = \Pr(\operatorname{sign}(\langle y, \alpha\rangle) \cdot \operatorname{sign}(\langle D_i, \alpha\rangle) = 1) = 1 - \frac{1}{\pi}\arccos\left(\sqrt{\frac{\operatorname{card}(D_i)}{\operatorname{card}(y)}}\right). \tag{17}$$

If D_i *is not included in* y, *i.e.*, $D_i \cap y = \emptyset$, *the signs of their random projections on a standard Gaussian random vector* α *follows the following joint distribution* $P2$:

$$P2(1) = \Pr(\operatorname{sign}(\langle y, \alpha\rangle) \cdot \operatorname{sign}(\langle D_i, \alpha\rangle) = -1) = \frac{1}{2}, \tag{18}$$

$$P2(2) = \Pr(\operatorname{sign}(\langle y, \alpha\rangle) \cdot \operatorname{sign}(\langle D_i, \alpha\rangle) = 1) = \frac{1}{2}. \tag{19}$$

Both $P1(l)$ *and* $P2(l)$ $(l = \{1,2\})$ *refer to the corresponding two probabilities associated with the two cases of the two random projection signs' product.*

Proof By substituting D_i and y into Theorem 1, the above distributions can be directly obtained. This completes the proof. Theorem 3 and the subsequent lemmas about the recovery bounds of CL are based on Theorem 1 and are independent of Theorem 2. □

Note that only the cardinality information of the label vector y is required in computing the joint distribution given in Theorem 3. Therefore, if we are given the distilled labelsets D

and y's cardinality, the joint distribution of an arbitrary D_i and y's random projection signs can be explicitly computed by using Theorem 3.

3.3 KL divergence test for recovery

Given classifiers W obtained in the training stage, the CL labels of a sample $x \in \mathbb{R}^p$ can be predicted as $z \in \{0,1\}^m$. Our goal in the recovery algorithm is to reconstruct the corresponding original label vector $y \in \{0,1\}^k$ from z. In CL, we propose a recovery algorithm via testing the KL divergence between possible joint distributions of random projection signs and the corresponding empirical joint distribution.

According to Theorem 3, given the cardinality of y's, the joint distribution of D_i and y's random projection signs can be quickly computed in two scenarios: D_i is included in y and D_i is not included in y. The corresponding empirical joint distribution $\hat{P}$ can be calculated from $v = \text{sign}(D_i A)$ and $z = \text{sign}(yA)$, i.e.,

$$\hat{P}(1) = \hat{\Pr}(\text{sign}(\langle y, \alpha\rangle) \cdot \text{sign}(\langle D_i, \alpha\rangle) = -1) = \frac{|j : z_j v_j = -1|}{m}, \tag{20}$$

$$\hat{P}(2) = \hat{\Pr}(\text{sign}(\langle y, \alpha\rangle) \cdot \text{sign}(\langle D_i, \alpha\rangle) = 1) = \frac{|j : z_j v_j = 1|}{m}. \tag{21}$$

The $\hat{P}(l)$ $(l = \{1,2\})$ refers to the corresponding two estimated probabilities associated with the two cases of the two random projection signs' product.

In the following discussion, the joint distribution associated with the situation that D_i is included in y is marked as $P1$, while the joint distribution associated with the situation that D_i is not included in y is marked as $P2$. Both $P1$ and $P2$ are defined in Theorem 3. We denote the empirical joint distribution computed from $v = \text{sign}(D_i A)$ and z as $\hat{P}$.

The target of recovery in CL is to determine whether D_i is included in y or not. This can be done by comparing the distance between $P1$ and $\hat{P}$ and that between $P2$ and $\hat{P}$. A smaller distance between $P1$ and $\hat{P}$ indicates a higher probability that D_i is included in y, while a smaller distance between $P2$ and $\hat{P}$ indicates a higher probability that D_i is not included in y.

KL divergence, also known as relative entropy, measures the distance between two probability distributions P and Q,

$$D_{KL}(P \| Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)}. \tag{22}$$

In the recovery algorithm of CL, given a y, we use KL divergence to measure the distance between $P1$ and $\hat{P}$ and the distance between $P2$ and $\hat{P}$ for different D_i. The differences between the two distances for all D_i are sorted. Then D_i with $\hat{P}$ closer to $P1$ than $P2$ are sequentially added into y as a subset from larger distance difference to smaller one until y reaching its cardinality. In particular, given the predicted CL label vector z of y, we calculate the following two KL divergences on each distilled labelset D_i in D:

$$M1_i = D_{KL}\left(P1 \| \hat{P}\right) = \sum_{j=1}^{2} P1(j) \log \frac{P1(j)}{\hat{P}(j)}, \tag{23}$$

$$M2_i = D_{KL}\left(P2 \| \hat{P}\right) = \sum_{j=1}^{2} P2(j) \log \frac{P2(j)}{\hat{P}(j)}. \tag{24}$$

The difference between $M1_i$ and $M2_i$ on each D_i forms a difference vector Diff:

$$\mathit{Diff} = M2 - M1, \quad \mathit{Diff} \in \mathbb{R}^d. \tag{25}$$

The positive entries of Diff corresponds to the distilled labelsets whose empirical joint distribution $\hat{P}$ is closer to $P1$ than $P2$, which indicates a higher possibility that the corresponding D_i are included in y. In addition, a larger and positive Diff_i implies D_i is more possible to be the subset of y than the other DLs with positive difference in Diff. In the recovery algorithm, we sort the positive entries of Diff from large to small, choose D_i sequentially from the ones with large Diff_i to the ones with small Diff_i, and add the selected D_i to y until the cardinality of y is arrived.

Since the cardinality of y cannot always be known previously in the prediction, we adds an outer loop to the above recovery procedure and searches the cardinality with the smallest recovery error $\|z - \mathrm{sign}(y_i A)\|$ in a given range. In CL, we choose it as the cardinality range of the training label vectors in Y.

Algorithm 2: Recovery algorithm of CL

Input: CL label vector $z \in \{-1, 1\}^m$, distilled labelsets D, Gaussian random matrix A used in compression, cardinality range $[\mathit{card1}, \mathit{card2}]$
Output: original label vector $y \in \{0, 1\}^k$
Calculate $V = \mathrm{sign}(DA)$ with $V_i = \mathrm{sign}(D_i A)$;
Calculate the joint distribution $P2$ by using (18) and (19);
for $i \leftarrow \mathit{card1}$ **to** $\mathit{card2}$ **do**
  for $j \in \{j : \mathrm{card}(D_j) \le i\}$ **do**
    Calculate the empirical joint distribution $\hat{P}_j$ by using (20) and (21) with $v = V_j$;
    Calculate the joint distribution $P1_j$ by using (16) and (17), wherein $\mathrm{card}(y) = i$;
    Calculate `KL divergence` between $P2$ and $\hat{P}_j$, i.e., $M2_j = \sum_{l=1}^{2} P2(l) \log \frac{P2(l)}{\hat{P}_j(l)}$;
    Calculate `KL divergence` between $P1_j$ and $\hat{P}_j$, i.e., $M1_j = \sum_{l=1}^{2} P1_j(l) \log \frac{P1_j(l)}{\hat{P}_j(l)}$;
    Calculate the difference between the two KL divergences $M2_j$ and $M1_j$, i.e., $\mathit{Diff}_j = M2_j - M1_j$;
  end
  Sort the positive entries in Diff from large to small;
  $y_i := \mathbf{0}$, $j := 1$;
  while $\mathrm{card}(y_i) < i$ **do**
    Add the distilled labelset D_j which is associated with the jth largest entry in sorted positive Diff to y_i, i.e., $y_i := y_i \cup D_j$;
  end
  Calculate the recovery error $\mathit{error}_i = \|z - \mathrm{sign}(y_i A)\|$;
end
Return $y = y_i$, wherein $i = \arg\min_i \mathit{error}_i$;

We show the recovery algorithm of CL in Algorithm 2 and highlight advantages of the recovery algorithm below

1. The recovery algorithm based on the KL divergence comparison only includes simple computations, e.g., comparison, sorting and thresholding. Thus the CL recovery algorithm is much faster than the normal compressed sensing algorithms used in Hsu et al. (2009).
2. Since the joint distribution of the random projection signs and the corresponding empirical distribution can be explicitly computed in the CL recovery algorithm, we can directly compare the distributions on their KL divergence rather than on their means or variances. This KL divergence comparison provides more useful information for testing whether a given DL is included in y. Compared with the mean test used in 1-bit compressed sensing, a direct comparison of distributions outputs more precise test result.
3. In the recovery algorithm, we test the existence of the distilled labelsets D rather than the single labels or random labelsets in y. This is an application of label dependence information in the multi-label prediction, because the distilled labelsets are the most frequent label subsets and the significant discrete patterns mining from the training label matrix. To see the benefits brought by this exploiting of label dependence, we compare the gaps between $P1$ and $P2$ in two situations, i.e., using distilled labelsets and using single labels in the test. Assume the dictionary corresponding to the single label situation is:

$$E = [E_1; E_2; \ldots; E_K], \quad E_i \in \{0,1\}^{n\times k}, \quad \text{card}\,(E_i) = 1. \tag{26}$$

According to the two joint distributions $P1$ and $P2$ given in Theorem 3, we can calculate the differences between $P1(l)$ and $P2(l)$ for all the $l = \{1, 2\}$:

$$\|P1(l) - P2(l)\| = \left\| \frac{1}{2} - \frac{1}{\pi} \arccos\left(\sqrt{\frac{\text{card}(D_i)}{\text{card}(y)}} \right) \right\|. \tag{27}$$

When dictionary generated by the single labels E is used, we have $\text{card}(D_i) = 1$ in (27). When distilled labelsets D is used instead, we have $\text{card}(D_i) > 1$ in (27). Thus the gap between $P1$ and $P2$ measured by $\|P1(l) - P2(l)\|$ is larger when distilled labelsets D is adopted. A larger gap between $P1$ and $P2$ will substantially reduce the number of failures in the tests of $\hat{P}$. Therefore, the recovery accuracy is improved in CL by using the label dependence information embedded in DL.

We show the training and prediction algorithms of CL in Algorithms 3 and 4, respectively.

3.4 Recovery bound

In order to investigate whether and when the CL recovery shown in Algorithm 2 is sufficiently accurate for reconstructing the original label vector y, we theoretically analyze the upper bounds for probabilities of 2 types of recovery failures, i.e.,

1. Type I failure: accepting labelset $b = D_i, \forall i = 1 : d$ included in y (i.e., $b \cap y = b$) when it is actually not (i.e., $b \cap y = \emptyset$). According to the recovery algorithm, this failure happens when the KL divergence between $P1$ and $\hat{P}$ is smaller than that between $P2$ and P, but P is actually estimated according to the samples from $P2$. The probability of this type of failure is

$$\Pr\left(D_{KL}\left(P1 \| \hat{P}\right) < D_{KL}\left(P2 \| \hat{P}\right)\right), \quad \text{when } \hat{P} = \hat{P2}, \text{ i.e., } b \cap y = \emptyset. \tag{28}$$

Algorithm 3: Training algorithm of CL

Input: Data matrix $X \in \mathbb{R}^{n\times p}$, label matrix $Y \in \{0,1\}^{n\times k}$, label compression dimension m, threshold τ, parameters for `SVM solver`
Output: m classifiers $W \in \mathbb{R}^{p\times m}$, distilled labelsets D, Gaussian random matrix A
label compression via random projections;
Generate a standard Gaussian random matrix $A \in \mathbb{R}^{k\times m}$;
Calculate the CL label matrix $Z = \text{sign}(YA)$;
classification via support vector machines;
for $i \leftarrow 1$ **to** m **do**
  Train binary classifier W_i on training set $\{X, Z_i\}$ by standard `SVM solver`, wherein Z_i is the ith column of Z;
end
$W = [W_1, W_2, \ldots, W_m]$;
distilled labelsets extraction;
Calculate the label powerset L, whose rows are unique label vectors in Y;
Extract the `Distilled labelsets (DL)` with input L and τ by using Algorithm 1;

Algorithm 4: Prediction algorithm of CL

Input: Sample $x \in \mathbb{R}^p$, CL classifiers $W \in \mathbb{R}^{p\times m}$, distilled labelsets D, Gaussian random matrix A used in compression, cardinality range $[card1, card2]$
Output: Label vector y
Calculate the predicted CL label vector $z \in \{-1,1\}^m$ of x by using CL classifiers W, i.e., $z = \text{sign}(xW)$;
Run `Recovery algorithm of CL` with input z, D, A and $[card1, card2]$ by using Algorithm 2;
Return the output y;

2. Type II failure: Excluding labelset $b = D_i, \forall i = 1:d$ from y (i.e., $b \cap y = \emptyset$) when it is actually included in y (i.e., $b \cap y = b$). According to the recovery algorithm, this failure happens when the KL divergence between $P1$ and $\hat{P}$ is larger than that between $P2$ and P, but P is actually estimated according to the samples from $P1$. The probability of this type of failure is

$$\Pr\left(D_{KL}\left(P1\|\hat{P}\right) > D_{KL}\left(P2\|\hat{P}\right)\right), \quad \text{when } \hat{P} = \hat{P}1, \text{ i.e., } b \cap y = b. \tag{29}$$

If the recovery algorithm is applied as a prediction model like in CL, from the view point of classification, the type I failure corresponds to *false positive* and its probability denotes 1-*specificity*, while the type 2 failure corresponds to *false negative* and its probability denotes 1-*sensitivity*. Thus, if the probabilities of the two types of failures can both be upper bounded by small probabilities, the prediction of CL will produce a satisfactory ROC (receiver operating characteristic).

In the following proofs, we derive the upper bounds for the probabilities of the two types of failures. Firstly, we show the difference between the 2 KL divergences is determined by $\hat{P}_1$, which is a binomial random variable. We give the distributions of $\hat{P}_1$ in the two cases

in Lemma 1. In Lemma 2, we study the conditions of $\hat{P}_1$ that lead to the two types of failures. Some significant properties of parameter δ in the conditions are provided in Lemma 3. By using the distributions of $\hat{P}_1$ and its conditions to cause the failures, we compute the probabilities of the failures in Proposition 1. Then the probabilistic bounds for the failures is derived in Theorem 4 based on Hoeffding's inequality. We analyze the improvement of measurement increase and distilled labelsets on the recovery bounds in Theorem 5.

For brevity of the analysis, we use the abbreviations $P_{1i} = P1(i)$, $P_{2i} = P2(i)$ and $\hat{P}_i = \hat{P}(i)$ for $i = 1, 2$. According to their definitions in (16)–(21), define

$$\gamma = P_{11} = \frac{1}{\pi} \arccos\left(\sqrt{\frac{\text{card}(b)}{\text{card}(y)}}\right) \in \left[0, \frac{1}{2}\right), \tag{30}$$

we calculate the difference between the 2 KL-divergences $D_{KL}(P1\|\hat{P})$ and $D_{KL}(P2\|\hat{P})$:

$$\begin{aligned}
&D_{KL}\left(P1\|\hat{P}\right) - D_{KL}\left(P2\|\hat{P}\right) \\
&= \sum_{i=1}^{2} P_{1i} \log \frac{P_{1i}}{\hat{P}_i} - \sum_{i=1}^{2} P_{2i} \log \frac{P_{2i}}{\hat{P}_i} \\
&= P_{11} \log \frac{P_{11}}{\hat{P}_1} + P_{12} \log \frac{P_{12}}{\hat{P}_2} - P_{21} \log \frac{P_{21}}{\hat{P}_1} - P_{22} \log \frac{P_{22}}{\hat{P}_2} \\
&= \gamma \log \frac{\gamma}{\hat{P}_1} + (1-\gamma) \log \frac{1-\gamma}{\hat{P}_2} - \frac{1}{2} \log \frac{1}{2\hat{P}_1} - \frac{1}{2} \log \frac{1}{2\hat{P}_2} \\
&= \left(\frac{1}{2} - \gamma\right) \log \frac{\hat{P}_1}{\hat{P}_2} + \gamma \log \gamma + (1-\gamma) \log (1-\gamma) + \log 2.
\end{aligned} \tag{31}$$

Therefore, the difference between the 2 KL-divergences is determined by the two variables $\hat{P}_1$ and $\hat{P}_2 = 1 - \hat{P}_1$, which distributions can be obtained by the following Lemma.

Lemma 1 *Given m measurements of random projection signs, the $m+1$ possible discrete values of $\{\hat{P}_1, \hat{P}_2\}$ are*

$$\hat{P}(j) = \left\{ \hat{P}_1 = \frac{j}{m}, \hat{P}_2 = \frac{m-j}{m} \right\}, \quad j = 0 : m, \tag{32}$$

where j is the number of -1s in the m values of $z_i v_i$ associated with $i = 1 : m$, while $m - j$ is the number of 1s. The probability of $\hat{P}(j)$ follows the following binomial distribution when $b \cap y = \emptyset$:

$$\Pr\left(\hat{P}(j)\right) = \binom{m}{j} \left(\frac{1}{2}\right)^m. \tag{33}$$

The probability of $\hat{P}(j)$ follows the following binomial distribution when $b \cap y = b$:

$$\Pr\left(\hat{P}(j)\right) = \binom{m}{j} (\gamma)^j (1-\gamma)^{m-j}. \tag{34}$$

Proof According to the definition of $\hat{P}$ in (20) and (21), $\hat{P}_1$ is the estimation for the probability of event $z_i v_i = -1$, while $\hat{P}_3 + \hat{P}_4$ is the estimation for the probability of event $z_i v_i = 1$. Thus the event $\hat{P}(j)$ is equivalent to

$$\hat{P}(j) = \{|i : z_i v_i = -1| = j, |i : z_i v_i = 1| = m - j\}, \quad j = 0 : m. \tag{35}$$

By using the results of Theorems 3 and 1, the distribution of random projection sign $z_i v_i$ can be obtained:

$$\begin{cases} \Pr(z_i v_i = -1) = \frac{1}{2}, \Pr(z_i v_i = 1) = \frac{1}{2}, & b \cap y = \emptyset; \\ \Pr(z_i v_i = -1) = \gamma, \Pr(z_i v_i = 1) = 1 - \gamma, & b \cap y = b. \end{cases} \tag{36}$$

Since the m values of $z_i v_i$ for different i are independent to each other, and the 2 distributions of each $z_i v_i$ under the two conditions are both Bernoulli distributions, $\hat{P}(j)$ follows the following binomial distribution:

$$\Pr\left(\hat{P}(j)\right) = \begin{cases} \binom{m}{j} \left(\frac{1}{2}\right)^j \left(\frac{1}{2}\right)^{m-j}, & b \cap y = \emptyset; \\ \binom{m}{j} (\gamma)^j (1-\gamma)^{m-j}, & b \cap y = b. \end{cases} \tag{37}$$

This leads to Lemma 1. □

By substituting (32) into (31), the difference between the 2 KL-divergences can be expressed as a function of j:

$$D_{KL}\left(P1 \| \hat{P}(j)\right) - D_{KL}\left(P2 \| \hat{P}(j)\right) = \left(\frac{1}{2} - \gamma\right) \log \frac{j}{m-j} + \gamma \log \gamma + (1-\gamma) \log (1-\gamma) + \log 2. \tag{38}$$

Therefore, given the original label vector y and a labelset $b \in D$, the sign of the difference between the 2 KL-divergences is determined by j. In the following lemma, we study which j will lead to the 2 types of failures.

Lemma 2 *When the probability estimation $\hat{P} = \hat{P}(j)$, with definition*

$$\delta = \left[\frac{1}{4\gamma(1-\gamma)}\right]^{\frac{1}{1-2\gamma}} \cdot \frac{\gamma}{1-\gamma}, \tag{39}$$

for KL-divergences $D_{KL}(P1 \| \hat{P}(j))$, $D_{KL}(P2 \| \hat{P}(j))$ and integer $j \in [0, m]$, we have

$$D_{KL}\left(P1 \| \hat{P}(j)\right) > D_{KL}\left(P2 \| \hat{P}(j)\right), \quad \forall j \in \left[\overline{\frac{m\delta}{1+\delta}}, m\right], \tag{40}$$

$$D_{KL}\left(P1 \| \hat{P}(j)\right) < D_{KL}\left(P2 \| \hat{P}(j)\right), \quad \forall j \in \left[0, \underline{\frac{m\delta}{1+\delta}}\right], \tag{41}$$

where $\overline{x}$ denotes the smallest integer larger than x and $\underline{x}$ denotes the largest integer smaller than x.

Proof According to (38), we have the following equivalences:

$$\begin{aligned}
&D_{KL}\left(P1\|\hat{P}(j)\right) > D_{KL}\left(P2\|\hat{P}(j)\right)\\
&\quad\Longleftrightarrow\quad D_{KL}\left(P1\|\hat{P}(j)\right) - D_{KL}\left(P2\|\hat{P}(j)\right) > 0\\
&\quad\Longleftrightarrow\quad \left(\frac{1}{2}-\gamma\right)\log\frac{j}{m-j} > -\gamma\log\gamma - (1-\gamma)\log(1-\gamma) - \log 2\\
&\quad\Longleftrightarrow\quad \log\frac{j}{m-j} > \log\left(\left[\frac{1}{4\gamma(1-\gamma)}\right]^{\frac{1}{1-2\gamma}}\cdot\frac{\gamma}{1-\gamma}\right)\\
&\quad\Longleftrightarrow\quad j > \frac{m\delta}{1+\delta}. \qquad (42)
\end{aligned}$$

Therefore, the necessary and sufficient condition for $D_{KL}(P1\|\hat{P}) > D_{KL}(P2\|\hat{P})$ is

$$j > \frac{m\delta}{1+\delta}. \qquad (43)$$

The same derivation leads to the necessary and sufficient condition for $D_{KL}(P1\|\hat{P}) < D_{KL}(P2\|\hat{P})$:

$$j < \frac{m\delta^2}{1+\delta^2}. \qquad (44)$$

Since j is an integer between 0 and m, these conditions lead to (40) and (41). This completes the proof. □

Before investigating the probabilities of the 2 types of failures, 5 significant properties of δ must be discussed.

Lemma 3 *The parameter δ defined in (39) has the following properties*:

$$\frac{\partial\delta}{\partial\gamma} > 0, \qquad (45)$$

$$\delta < 1, \qquad (46)$$

$$\frac{\delta}{1+\delta} - \frac{1}{2} < 0, \qquad (47)$$

$$\frac{\delta}{1+\delta} - \gamma > 0, \qquad (48)$$

$$\frac{\partial\left(\frac{\delta}{1+\delta}-\gamma\right)}{\partial\gamma} < 0. \qquad (49)$$

Proof Since δ is a function of γ, its derivative can be calculated according to fundamental differentiation rules:

$$\frac{\partial\delta}{\partial\gamma} = \frac{\partial\left(\frac{1}{4\gamma(1-\gamma)}\right)^{\frac{1}{1-2\gamma}}}{\partial\gamma}\cdot\frac{\gamma}{1-\gamma} + \left(\frac{1}{4\gamma(1-\gamma)}\right)^{\frac{1}{1-2\gamma}}\cdot\frac{\partial\frac{\gamma}{1-\gamma}}{\partial\gamma}. \qquad (50)$$

Define function φ as

$$\varphi = \left(\frac{1}{4\gamma\,(1-\gamma)}\right)^{\frac{1}{1-2\gamma}}. \tag{51}$$

Then its derivative can be calculated by computing the logarithm of both sides:

$$\ln\varphi = \frac{1}{1-2\gamma}\ln\frac{1}{4\gamma\,(1-\gamma)}. \tag{52}$$

Computing the derivatives of both sides yields

$$\frac{\varphi'}{\varphi} = \frac{2}{(1-2\gamma)^2}\ln\frac{1}{4\gamma\,(1-\gamma)} - \frac{1}{\gamma\,(1-\gamma)}. \tag{53}$$

Thus we have

$$\varphi' = \frac{\partial(\frac{1}{4\gamma(1-\gamma)})^{\frac{1}{1-2\gamma}}}{\partial\gamma} = \frac{2\varphi}{(1-2\gamma)^2}\ln\frac{1}{4\gamma(1-\gamma)} - \frac{\varphi}{\gamma(1-\gamma)}. \tag{54}$$

By substituting (51) into (47) and using the definition of φ, we obtain

$$\frac{\partial\delta}{\partial\gamma} = \frac{2\gamma\varphi}{(1-\gamma)(1-2\gamma)^2}\ln\frac{1}{4\gamma(1-\gamma)}. \tag{55}$$

The inequality $\gamma \in [0, 1/2)$ yields

$$1-\gamma > 0, \qquad \ln\frac{1}{4\gamma\,(1-\gamma)} > 0, \qquad \varphi > 0. \tag{56}$$

These lead to $\frac{\partial\delta}{\partial\gamma} > 0$, which completes the proof of (45).

Since $\frac{\partial\delta}{\partial\gamma} > 0$ and $\gamma \in [0, 1/2)$, we have

$$\begin{aligned}\log\delta &< \lim_{\gamma\to 1/2}\log\delta = \lim_{\gamma\to 1/2}\left[\frac{1}{1-2\gamma}\log\frac{1}{4\gamma\,(1-\gamma)} + \log\frac{\gamma}{1-\gamma}\right]\\ &= \lim_{\gamma\to 1/2}\left[\frac{-\log 4\gamma\,(1-\gamma)}{1-2\gamma}\right] = \lim_{\gamma\to 1/2}\left[\frac{-\partial\log 4\gamma\,(1-\gamma)}{\partial\,(1-2\gamma)}\right]\\ &= \frac{1-2\gamma}{2\gamma\,(1-2\gamma)} = 0.\end{aligned} \tag{57}$$

The monotonicity of logarithm and $\log\delta < 0$ yield $\delta < 1$. This completes the proof of (46).

By using (55), we have

$$\begin{aligned}\frac{\partial\frac{\delta}{1+\delta}}{\partial\gamma} &= \frac{\partial\frac{\delta}{1+\delta}}{\partial\delta}\cdot\frac{\partial\delta}{\partial\gamma}\\ &= \frac{1}{(1+\delta)^2}\cdot\frac{2\gamma\varphi}{(1-\gamma)\,(1-2\gamma)^2}\ln\frac{1}{4\gamma\,(1-\gamma)} &(58)\\ &= \frac{2\delta}{(1+\delta)^2\,(1-2\gamma)^2}\ln\frac{1}{4\gamma\,(1-\gamma)} > 0. &(59)\end{aligned}$$

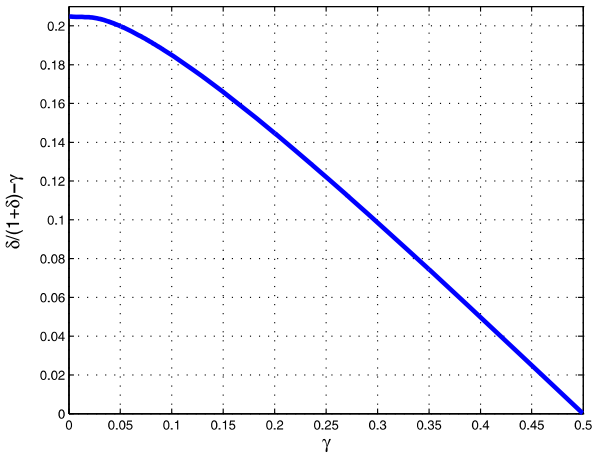

Fig. 3 Plot of $\delta/(1+\delta)-\gamma$ as a function of $\gamma \in [0, 1/2)$ on 5000 points between 0 and 1/2

Since $\gamma \in [0, 1/2)$, we have

$$\frac{\delta}{1+\delta} < \left.\frac{\delta}{1+\delta}\right|_{\gamma=1/2} = \frac{1}{2}. \tag{60}$$

This completes the proof of (47).

It is tedious and unnecessary to prove inequalities (48) and (49) via similar derivation to above ones, because very high (more than 6th) order derivatives of $(\delta/(1+\delta)-\gamma)^2$ have to be computed in this case, while this function is too complex to compute its high order derivatives. Hence we prove (48) and (49) by plotting $\delta/(1+\delta)-\gamma$ as a function of $\gamma \in [0, 1/2)$ in Fig. 3.

According to the curve of the function $\delta/(1+\delta)-\gamma$ shown in Fig. 3, the function is always larger than 0 and decreases as γ increasing in interval $\gamma \in [0, 1/2)$. Since $\delta/(1+\delta^2)-\gamma$ is a continues function of γ, there is no spike or singularity on its curve. Therefore, we obtain (48) and (49). This completes the proof. □

According to Lemmas 1 and 2, we have the following proposition about the probabilities of the 2 types of recovery failures.

Proposition 1 *The probabilities of the* 2 *types of recovery failures are functions of m and* γ:

$$\Pr\left(D_{KL}\left(P1\|\hat{P}\right) < D_{KL}\left(P2\|\hat{P}\right)\right) = \sum_{j=1}^{\frac{m\delta}{1+\delta}} \binom{m}{j}\left(\frac{1}{2}\right)^m, \quad b \cap y = \emptyset, \tag{61}$$

$$\Pr\left(D_{KL}\left(P1\|\hat{P}\right) > D_{KL}\left(P2\|\hat{P}\right)\right) = \sum_{j=\frac{m\delta}{1+\delta}}^{m} \binom{m}{j}\gamma^j(1-\gamma)^{m-j}, \quad b \cap y = b. \tag{62}$$

Proof Lemma 1 gives the distribution of $\hat{P}(j)$ follows binomial models (33) and (34) respectively under the 2 different facts, i.e., $b \cap y = \emptyset$ in type I failure and $b \cap y = b$ in type II failure. Lemma 2 provides the 2 ranges of j that leads to the 2 kinds of failures in (40) and

(41), respectively. Therefore, the probability of each kind of failure is the sum of all the probabilities $\hat{P}(j)$ with j in the respective range. Thus we have

$$\Pr\left(D_{KL}\left(P1\|\hat{P}\right) < D_{KL}\left(P2\|\hat{P}\right)\right) = \sum_{j=1}^{\frac{m\delta}{1+\delta}} \Pr\left(\hat{P}(j)\right), \quad b \cap y = \emptyset, \tag{63}$$

$$\Pr\left(D_{KL}\left(P1\|\hat{P}\right) > D_{KL}\left(P2\|\hat{P}\right)\right) = \sum_{j=\frac{m\delta}{1+\delta}}^{m} \Pr\left(\hat{P}(j)\right), \quad b \cap y = b. \tag{64}$$

Substitute $\Pr(\hat{P}(j))$ given in Lemma 1 into the above probabilities, we obtain (61) and (62) in Proposition 1. This completes the proof. □

Since $\hat{P}(j)$ follows binomial distribution and the probabilities of the 2 kinds of failures are CDFs of binomial distributions, we apply Hoeffding's inequality to the 2 probabilities in Proposition 1 and obtain the upper bounds for them.

Theorem 4 (Probabilistic bounds for recovery failures) *The upper bounds for the probabilities of the two types of recovery failures are*:

$$\Pr\left(D_{KL}\left(P1\|\hat{P}\right) < D_{KL}\left(P2\|\hat{P}\right)\right) \le \frac{1}{2}\exp\left[-2\left(\frac{\delta}{1+\delta} - \frac{1}{2}\right)^2 m\right], \quad b \cap y = \emptyset, \tag{65}$$

$$\Pr\left(D_{KL}\left(P1\|\hat{P}\right) > D_{KL}\left(P2\|\hat{P}\right)\right) \le \frac{1}{2}\exp\left[-2\left(\frac{\delta}{1+\delta} - \gamma\right)^2 m\right], \quad b \cap y = b. \tag{66}$$

Proof For type I failure, its probability (61) is a CDF of a binomial distribution (33). According to the condition of Hoeffding's inequality

$$\frac{m\delta}{1+\delta} \le \frac{m\delta}{1+\delta} < \frac{1}{2} \cdot m, \tag{67}$$

we apply Hoeffding's inequality to (61) and obtain

$$\begin{aligned}\Pr\left(D_{KL}\left(P1\|\hat{P}\right) < D_{KL}\left(P2\|\hat{P}\right)\right) &= \sum_{j=1}^{\frac{m\delta}{1+\delta}} \binom{m}{j}\left(\frac{1}{2}\right)^m \\ &\le \frac{1}{2}\exp\left[-2\frac{(\frac{m\delta}{1+\delta} - \frac{m}{2})^2}{m}\right] \\ &\le \frac{1}{2}\exp\left[-2\left(\frac{\delta}{1+\delta} - \frac{1}{2}\right)^2 m\right], \quad b \cap y = \emptyset. \end{aligned} \tag{68}$$

The second inequality in the above derivation is due to (47) in Lemma 3, which yields

$$\frac{m\delta}{1+\delta} - \frac{m}{2} \le \left(\frac{\delta}{1+\delta} - \frac{1}{2}\right) m < 0. \tag{69}$$

This completes the proof of (65).

For type II failure, its probability (62) can be written as a CDF of a binomial distribution related to (34):

$$\Pr\left(D_{KL}\left(P1\|\hat{P}\right) > D_{KL}\left(P2\|\hat{P}\right)\right) = \sum_{j=\overline{\frac{m\delta}{1+\delta}}}^{m} \binom{m}{j} \gamma^j (1-\gamma)^{m-j}$$
$$= \sum_{j=0}^{m-\overline{\frac{m\delta}{1+\delta}}} \binom{m}{j} (1-\gamma)^j \gamma^{m-j}. \tag{70}$$

The inequality (48) in Lemma 3 yields the condition of Hoeffding's inequality.

$$m - \overline{\frac{m\delta}{1+\delta}} \leq m - \frac{m\delta}{1+\delta} < (1-\gamma)\cdot m. \tag{71}$$

We apply Hoeffding's inequality to (70) and obtain

$$\Pr\left(D_{KL}\left(P1\|\hat{P}\right) > D_{KL}\left(P2\|\hat{P}\right)\right)$$
$$\leq \frac{1}{2}\exp\left[-2\frac{\left[\left(m - \overline{\frac{m\delta}{1+\delta}}\right) - m(1-\gamma)\right]^2}{m}\right]$$
$$\leq \frac{1}{2}\exp\left[-2\left(\frac{\delta}{1+\delta} - \gamma\right)^2 m\right], \quad b \cap y = b. \tag{72}$$

The second inequality in the above derivation is due to (48) in Lemma 3, which yields

$$\left(m - \overline{\frac{m\delta}{1+\delta}}\right) - m(1-\gamma) = m\gamma - \overline{\frac{m\delta}{1+\delta}} \leq \left(\gamma - \frac{\delta}{1+\delta}\right) m < 0. \tag{73}$$

This completes the proof of (66). □

The probabilistic bounds given in Theorem 4 are exponential functions of m and γ, wherein m is the number of measurements (number of random projection signs in the recovery algorithm, i.e., dimension of CL label vector in CL), and $\gamma \in [0, 1/2)$ is a monotonically decreasing function of card(b) according to (36). As a compression-recovery algorithm, it is essential to investigate how many measurements are sufficient to ensure the success of the recovery in CL. As a multi-label prediction algorithm, it is essential to analyze whether the application of distilled labelsets (DLs) can improve the prediction performance. These two significant questions can be well answered by Theorem 5 that analyzes how the probabilistic bounds of recovery failures change with m and γ.

Theorem 5

1. *The upper bounds of the probabilities for the* 2 *types of recovery failures exponentially shrink with the increasing of measurements' number* m.
2. *The upper bounds of the probabilities for the* 2 *types of recovery failures exponentially shrink with the increasing of the distilled labelset* b*'s cardinality.*

Proof 1 In Theorem 4, the two upper bounds

$$\frac{1}{2}\exp\left[-2\left(\frac{\delta}{1+\delta}-\frac{1}{2}\right)^2 m\right] \quad \text{and} \quad \frac{1}{2}\exp\left[-2\left(\frac{\delta}{1+\delta}-\gamma\right)^2 m\right]. \tag{74}$$

They are both exponentials of negative linear functions of m. Therefore, they exponentially decreases with the increase in the number of measurements m. This completes the proof of conclusion 1.

2 We study the monotonicities of the 2 functions of γ in the 2 upper bounds from Theorem 4:

$$f_1(\gamma)=\left(\frac{\delta}{1+\delta}-\frac{1}{2}\right)^2 \quad \text{and} \quad f_2(\gamma)\left(\frac{\delta}{1+\delta}-\gamma\right)^2. \tag{75}$$

Their partial derivatives with respect to γ are

$$\frac{\partial f_1}{\partial \gamma}=2\left(\frac{\delta}{1+\delta}-\frac{1}{2}\right)\cdot\frac{\partial \frac{\delta}{1+\delta}}{\partial \gamma}, \tag{76}$$

$$\frac{\partial f_2}{\partial \gamma}=2\left(\frac{\delta}{1+\delta}-\gamma\right)\cdot\frac{\partial(\frac{\delta}{1+\delta}-\gamma)}{\partial \gamma}. \tag{77}$$

By using inequalities (47), (45), (48), (49) and (59), we have

$$\frac{\partial f_1}{\partial \gamma}<0 \quad \text{and} \quad \frac{\partial f_2}{\partial \gamma}<0. \tag{78}$$

Thus f_1 and f_2 increase with the decreasing of γ. Since both f_1 and f_2 are nonnegative, the 2 upper bounds

$$\frac{1}{2}\exp\left[-2f_1(\gamma)m\right] \quad \text{and} \quad \frac{1}{2}\exp\left[-2f_2(\gamma)m\right] \tag{79}$$

exponentially decrease with the decreasing of γ.

According to (36), $\gamma\in[0,1/2)$ is a monotonically decreasing function of card(b), so the 2 upper bounds in Theorem 4 exponentially decreases with the increase in the cardinality of the DL b. This completes the proof of conclusion 2. □

Theorem 5 shows that the upper bounds for the probabilities of the 2 types of failures both the recovery accuracy exponentially shrink with the increasing of either the measurements or the cardinality of DL. It indicates that the multi-label prediction performance of CL will not be harmed by the label compression and will be significantly improved by applying LDM. In summary, Theorem 5 This theoretically shows CL saves the time costs and balances the training data, and LDM explores the label correlations. When DL is replaced with unit vectors whose elements are 0 except one element is 1, it is easy to derive from Theorem 5 that $m=\mathcal{O}(\log k)$ can produce a sufficiently accurate recovery. Note that DL can produce more robust and accurate recovery than the unit vectors (cf. (27) and the analysis below it), though the value of m ensuring accurate recovery cannot be precisely identified in this case.

4 Discussion

In this section, we discuss CL's contributions to multi-label learning, and analyze CL's relationships with compressed sensing (CS) and error-correcting output codes (ECOC).

4.1 Contributions to multi-label learning

In essence, CL is a label transformation method for multi-label learning. It solves or at least substantially alleviates the three aforementioned problems harassing multi-label learning field via label random projection based compression and recovery. In particular, the label compression in CL generates a new label matrix with improved sample balance for each label, mutual independence between different labels and lower label dimensionality than the original one. According to the analyses in Sect. 2, these are attributed to a series of properties of random projection signs. The sample balance and mutual independence of CL labels remove the two obstacles of applying conventional binary classification methods to multi-label learning problems. Thus the label compression method in CL avoids the problems of directly applying single-label learning methods to multi-label learning tasks, while inherits their advantages in binary classification tasks. Besides, the single-label learning methods are directly invoked in CL without any modification. For example, the SVM based CL presented in this paper retains the optimality and robustness of margin maximization, and can be extended to nonlinear kernel space. The dimension reduction of the label matrix significantly decreases the problem size and yields an efficient training stage. These characteristics of CL improves both effectiveness and efficiency of multi-label learning.

Although the three problems have been more or less considered by existing methods, they have rarely been simultaneously considered without introducing other problems. For instance, methods for exploiting the label dependence usually expand the problem size and aggravate the sample imbalance. Methods for label dimension reduction often transform the classification problem to other problems that ignore the label dependence and have complex label recovery algorithms.

Moreover, CL is a general method whose training process is isolated from the label compression and recovery, and thus various existing single-label and multi-label learning methods can be directly invoked in the training stage of CL by using the compressed label matrix. The benefits of these methods can be completely retained in their CL variants. Hence CL is not only a multi-label classification solver but also a general method that can be directly applied to most existing multi-label leaning techniques for improving their performance and speed. LDM in CL is a greedy binary matrix decomposition technique that exploits the discrete patterns of a given binary matrix. Thus LDM can be isolated from the CL scheme and applied independently to multi-label learning in order to obtain better labelsets for LP (Tsoumakas and Katakis 2007) or build the nodes in tree-structural hierarchy (Bianchi et al. 2006; Tsoumakas et al. 2008).

Furthermore, CL significantly decreases the problem size of multi-label learning by adding a label compression and recovery procedure. However, the computation of the additional procedure is simple and fast. In particular, the compression is composed of random projections and hard thresholding, and the recovery includes only comparison, thresholding and sorting. Thus the additional time cost brought by compression is negligible comparing with the tremendous reduced time cost caused by CL. Compared with the multi-label learning via CS (Hsu et al. 2009) whose recovery needs to solve an ℓ_1 minimization, CL targets on a more powerful classification model and develops a more efficient recovery algorithm by exploiting the $\{0, 1\}$ nature and discrete pattern of the label matrix. Therefore, CL brings significant acceleration to multi-label learning and makes the large-scale problems (Koufakou et al. 2011) computationally tractable.

4.2 Relationship with compressed sensing

CS (Donoho 2006; Candès et al. 2006) is a sparse signal compression and recovery scheme that achieves remarkable success in recent years. It proves that a sparse signal x can be exactly recovered from a small number of its random projections $y = xA$ if the projection matrix A follows the Restricted Isometry Property (RIP). Similarly, CL proposes a $\{0, 1\}$ label matrix compression and recovery scheme. Thus it is interesting to discuss the relationship between CS and CL.

1. CS and CL both use random projections in their compression of a sparse signal and a $\{0, 1\}$ label matrix, respectively. Random projection can be deemed as a pseudorandom generator that encodes certain information of the original data into random variables. In CS, random projections provide linear random measurements that satisfy RIP, which guarantees the success of exact recovery of sparse signal x from a few of its random measurements y via ℓ_1 minimization. In CL, random projection signs of two binary vectors have an explicit joint distribution, so a row of $\{0, 1\}$ label matrix can be exactly recovered from its random projection signs by testing the joint distributions of the random projection signs of the row and several given $\{0, 1\}$ vectors, i.e., distilled labelsets. Note that the measurements of CS are real-valued while the measurements (i.e., CL labels) are 1-bit. From the viewpoint of transmission, the space costs of CL compression is significantly reduced comparing with the CS compression.
2. CS and CL develop different recovery algorithms by exploiting the different properties of sparse signals and the $\{0, 1\}$ label matrix, respectively. In CS, the sparsity leads to the minimization of ℓ_1 norm of the unknown signal in the recovery algorithm, because ℓ_1 norm is a convex relaxation of the signal cardinality. In CL, although the rows of the $\{0, 1\}$ label matrix satisfy the sparse assumption as well, they have more specific characteristics. In particular, (1) each entry is either 0 or 1 in value; and (2) there exist discrete patterns that are frequently shared by the rows of the label matrix. These two characteristics of the $\{0, 1\}$ label matrix inspire the KL divergence based hypothesis tests in the recovery algorithm of CL.

A more related CS problem to the CL problem is 1-bit compressed sensing (Gupta et al. 2010). Different from CS which recovers the sparse signal from a few of its random projections, 1-bit CS aims to recover the support set of the sparse signal x from a few of its random projection signs $y = \text{sign}(xA)$. In the passive algorithm of 1-bit CS, the expectation of $y_i \text{sign}(A_{ji})$ over different i can be explicitly computed in a similar spirit of the geometric inference in Theorem 1. A hypothesis test of the expectation is then conducted to determine whether j is included in the support set of x.

The similarity between 1-bit CS and CL is as follows: the geometric inference of the expectation in 1-bit CS and the geometric inference of the joint distribution in CL both study the properties of random projection on a vector drawn uniformly on a hypersphere.

However, 1-bit CS and CL are different in the following three aspects:

1. Their problems are different. 1-bit CS targets on recovering the support set of a sparse signal, while CL aims at recovering a $\{0, 1\}$ label matrix. Recovering the support set in 1-bit CS only cares about the positions of the nonzero entries in a single sparse vector, while recovering $\{0, 1\}$ label matrix in CL explores the frequent $\{0, 1\}$ patterns appearing in all the rows of the matrix via LDM.
2. 1-bit CS studies the random variable $y_i \text{sign}(A_{ji})$, which can be explained as the product of random projection signs of sparse signal x and a unit vector e with 1 on the jth entry

and 0 otherwise. CL studies the random projection signs of $\{0, 1\}$ label vector y and a given distilled labelsets D_i.

3. 1-bit CS recovers the support set through tests of the expectation of two random projection signs' product, while CL recovers the $\{0, 1\}$ label matrix via tests of the joint distribution of two random projection signs based on KL divergence comparison.

Although 1-bit CS and CL are two different techniques for two different problems, it is possible to extend the methods used in CL to 1-bit CS for improving its performance and recovery bound. In particular, the expectation of $y_i \text{sign}(A_{ji})$ used in the 1-bit CS recovery can be replaced by $y_i \text{sign}(zA_i)$, wherein z is a vector revealing the structure information of the original signal x. This will lead to a "structured 1-bit CS". Another modification from CL is to adopt a distribution test based on KL divergence rather than an expectation test in the recovery of 1-bit CS, because the distribution includes more information about x than its expectation.

4.3 Relationship with error-correcting output codes

Error-correcting output codes (ECOC) (Dietterich and Bakiri 1995; Escalera et al. 2010) transforms multi-class problem to several binary problems. ECOC consists of two stages, (1) coding stage that encodes each class label into a $\{-1, +1\}$ or $\{-1, 0, +1\}$ codeword, and (2) decoding stage that seeks for the class codeword closest to the predicted codeword of a test sample. Each codeword of the training sample is a $\{-1, +1\}$ or $\{-1, 0, +1\}$ vector of dimension d that augments with the increasing of the class numbers k. In the training step, the codewords of training samples are deemed as their new label vectors, and d binary classifiers are learned from the training set to predict the d dimensions of the codeword of a given sample, respectively. It has been shown that the error-correcting properties of the decoding stage are helpful to reduce the bias and variance of the learning algorithm. Please refer to Escalera et al. (2010) for a complete review of ECOC methods.

ECOC is related to CL for the following two reasons:

1. In label compression via random projections, CL transforms the original label matrix into a new one, i.e., the CL label matrix. This compression step is similar to the coding stage of ECOC, which assigns each class a new label vector called codeword. Both CL labels and codewords in ECOC are used as the labels of the training samples when training the subsequent binary classifiers.
2. In the recovery by KL-divergence test, CL recovers the original label vector of a test sample from its CL label vector predicted by the binary classifiers. This recovery step is similar to the decoding stage of ECOC, which finds the class codeword closest to the predicted codeword of a test sample and assigns the corresponding class label to the sample. The final prediction results of CL and ECOC are inferred from the predicted CL label vector and ECOC codeword, respectively.

Although CL has a coding stage and a decoding stage similar to the two stages in ECOC, these two schemes are essentially different on their targeted problems, coding and decoding algorithms, and methodologies. Details are given below.

1. ECOC is designed for a multi-class problem, while CL is designed for a multi-label learning problem. Although a multi-class problem can be viewed as a special case of multi-label learning, their discrepancy leads to the key differences in developing ECOC and CL. For example, there are only k possible codewords for a multi-class problem in ECOC, so the training set can include all the k codewords for most datasets. Thus these k

codewords can be independently generated in the coding stage, and treated isolatedly in the training and decoding stages. However, there are at most $2^k - 1$ possible label vectors (exponentially increased with k) for a multi-label learning problem in CL, so the training sets in most datasets cannot include all the possible CL label vectors. This fact indicates the label vector that needs to be predicted could never appear in the training set before. For this reason, the correlation between different label vectors and the dependence between different labels play significantly roles in multi-label learning and provide critical information for prediction. Thus the CL label vectors cannot be generated independently in compression. This is why we apply the same random matrix A to different label vectors in CL. This is also why we develop LDM to extract the label correlations, preserve them in DL and use DL in the CL recovery. Another difference caused by the problem discrepancy is the decoding stage. It is possible to search for the closest codeword among the k possible ones in ECOC. But it is impossible to search for the most accordable label vector among the $2^k - 1$ possible ones in CL, at least from the aspect of computation. Thus we have to develop an accurate recovery algorithm for finding the real label vector in CL.

2. The coding stage in ECOC and the label compression in CL are different in the dimension of a codeword (the number of CL labels) and the correlation preservation. Common coding strategies in ECOC include one-versus-all, one-versus-one, randomized design such as dense random and sparse random, and problem-dependent design such as DECOC, Forest-ECOC and ECOC-ONE. One-versus-all and one versus-one suffer from the problem of serious imbalance in the coding matrix whose rows are composed of codewords. Moreover, one-versus-one needs to train $k(k-1)/2$ binary classifiers in the subsequent training step, which is computationally intractable for most datasets. Randomized design generates the codeword for each class randomly and independently. The burden brought by randomness is that high dimensionality of the codeword is required to preserve sufficient information of the data. Problem-dependent design extracts the codewords by mining the structure of the k classes. This method requires codewords of dimension at least $k-1$. However, the imbalance problem of the coding matrix is ignored in this method. In summary, the dimensionality of codewords in ECOC is much higher than or at least around k in most strategies to guarantee successful coding, and the correlations between different classes can be abandoned after coding. However, CL can compress the k-dimensional label vector to $m = \mathcal{O}(\log k)$ (cf. the last paragraph of Sect. 3.4) in order to solve the problem of label high dimensionality. Moreover, the distribution and pairwise distance of the original label vectors are maintained after compression in CL (cf. the last paragraph of Sect. 2.3). The preserved graph structure is helpful for training classifiers on the CL labels. Note CL adopts random projection signs as the compression of the original labels, which has not been used in the coding method of ECOC.
3. The decoding stage of ECOC and the KL divergence test based recovery in CL are different in algorithm development. The decoding stage in ECOC is based on the error-correcting principles and finds the closest codeword among the k possible ones, while the recovery of CL precisely reconstructs the original label vector dimension-wisely or DL-wisely (by testing whether each DL belongs to the original label vector or not). The decoding algorithm of ECOC can be grouped into three types, i.e., comparison of distances between the predicted codeword and existing ones, class membership possibility estimation and pattern space transformation. But the recovery of CL is based on a series of KL divergence tests on DL.

In ML-CS (Hsu et al. 2009), the regret transform in SECOC (Langford and Beygelzimer 2005) (one method of ECOC) provides a theoretical guarantee similar in formulation

to RIP in compressed sensing. Therefore, to some extent, ML-CS (Hsu et al. 2009) can be deemed as an extension of ECOC in multi-label learning. Compared with this method, CL performs promisingly on three aspects: (1) less measurements, ML-CS needs $m = \mathcal{O}(\log k)$ real-valued labels, while CL requires only $m = \mathcal{O}(\log k)$ 0-1 (1-bit) labels; (2) faster recovery, ML-CS invokes existing convex optimization based CS recovery algorithms, while CL develops a non-iterative recovery algorithm with linear time; (3) ML-CS transforms a 0-1 prediction problem to a regression problem, while CL transforms a large 0-1 prediction problem to a smaller one. Thus CL does not change the nature of the problem; (4) the associated improvement on prediction performance by exploring label correlation in ML-CS cannot be analyzed, while CL thoroughly explores the label correlation via developing LDM and applying DL in recovery. The benefits of applying DL can be clearly analysed (cf. the last two paragraphs in Sect. 3.3).

In addition, CL can be applied to a multi-class problem without the application of LDM and DL. Comparing with the existing ECOC methods, CL brings the following advantages for multi-class problems: (1) smaller dimensionality of the codewords; (2) randomized coding that can simultaneously preserve class correlations and eliminate imbalance in the coding matrix; and (3) more robust and faster decoding based on an accurate recovery. Therefore, CL on unit vectors can be deemed as a powerful and novel ECOC method for multi-class problems.

5 Experiments

In this section, we evaluate CL via 5 groups of experiments on 21 datasets obtained from real-world problems including text classification, image annotation, scene classification, music categorization, genomics and web page classification. In the first group of experiments, we evaluate the label compression and recovery in CL by measuring the sample balance, mutual independence of CL label matrix Z and the recovery error rate of Algorithm 2. In the second group of experiments, we compare CL with BR in multi-label prediction when SVM is adopted as the binary classifier in both methods. Time cost and prediction performance are evaluated on different C parameters of SVM for a comparison of robustness. In the third group of experiments, we compare CL with 3 popular multi-label learning methods, i.e., ML-knn (Zhang and Zhou 2007), MDDM (Zhang and Zhou 2008) and multi-label prediction via compressed sensing (ML-CS) (Hsu et al. 2009). ML-knn and MDDM are extensions of knn and dimension reduction (Si et al. 2010; Guan et al. 2011) for multi-label learning, respectively. ML-CS is related to CL because it adopts a compressed label embedding. In the fourth group of experiments, we compare CL with 2 SVM methods dealing with imbalanced data, i.e., SVM-SMOTE (Chawla et al. 2002) and SVM-WEIGHT (Osuna et al. 1997). This group of experiments separates the performance improvements caused by label correlation and sample balance in CL. In the fifth group of experiments, we study the trade-off between time cost reduced by label compression and prediction performance enhanced by sample balance in CL. In order to eliminate the influence of label correlation, DL is replaced by the dictionary composed of unit vectors. In the last 4 groups of experiments, the performance of multi-label prediction is evaluated in terms of 5 metrics, i.e., Hamming loss, precision, recall, F1 score and accuracy, and CPU seconds for time cost contest. All the experiments are implemented and run in MatLab on a server with dual quad-core 3.33 GHz Intel Xeon processors and 32 GB RAM.

5.1 Evaluation metrics

In the experiments of label compression and recovery, three metrics are used to measure the sample balance, label independence and recovery error rate, respectively. Given a $\{0, 1\}$ label matrix $Y \in \{0, 1\}^{n\times k}$, its balance-of-degree is defined as the average proportion of positive samples on all the different labels in (2). A degree-of-balance close to 0.5 indicates a good sample balance in Y.

The label independence in Y is measured by the average χ^2 score after Yates' correction (Yates 1934) over all the label pairs in Y. In particular, for a label pair $\{i, j\}$, assume

$$a = \left|p : Y_{pi} = 1, Y_{pj} = 1\right|, \qquad b = \left|p : Y_{pi} = -1, Y_{pj} = 1\right|, \tag{80}$$

$$c = \left|p : Y_{pi} = 1, Y_{pj} = -1\right|, \qquad d = \left|p : Y_{pi} = -1, Y_{pj} = -1\right|. \tag{81}$$

The χ^2 score for label pair $\{i, j\}$ is defined as:

$$\chi_{ij}^2 = \frac{n\left(\|ad - bc\|_1 - n/2\right)^2}{(a+b)(c+d)(b+d)(a+c)}. \tag{82}$$

The average χ^2 score is:

$$\chi^2 = \text{mean}\left(\chi_{ij}^2\right). \tag{83}$$

A large χ^2 score indicates a strong mutual independence between labels.

Given two label matrices $Y1, Y2 \in \{0, 1\}^{n\times k}$, wherein $Y1$ is the real one an $Y2$ is the recovered one, the recovery error rate in CL is measured by the Hamming loss:

$$HamLoss = \frac{1}{nk}\sum_{i=1}^{n}\sum_{j=1}^{k} Y1_{ij} \oplus Y2_{ij}, \tag{84}$$

where $\oplus$ is the exclusive disjunction, i.e., the XOR operation.

In the experiments of multi-label prediction, five metrics, i.e., Hamming loss, precision, recall, F1 score and accuracy, are used to measure the prediction performance. Hamming loss is defined in (84). The other four metrics are defined as:

$$Precision = \frac{1}{n}\sum_{i=1}^{n}\frac{\text{card}\left(Y1_i \cap Y2_i\right)}{\text{card}\left(Y2_i\right)}, \tag{85}$$

$$Recall = \frac{1}{n}\sum_{i=1}^{n}\frac{\text{card}\left(Y1_i \cap Y2_i\right)}{\text{card}\left(Y1_i\right)}, \tag{86}$$

$$F1 = \frac{1}{n}\sum_{i=1}^{n}\frac{2\text{card}\left(Y1_i \cap Y2_i\right)}{\text{card}\left(Y1_i\right) + \text{card}\left(Y2_i\right)}, \tag{87}$$

$$Accuracy = \frac{1}{n}\sum_{i=1}^{n}\frac{\text{card}\left(Y1_i \cap Y2_i\right)}{\text{card}\left(Y1_i \cup Y2_i\right)}. \tag{88}$$

These five metrics have been broadly applied on general binary data. However, their importances differs when used in multi-label prediction evaluation, because there are much more 1s than 0s in the label matrix. Hamming loss could be very small when the labels of all

Table 1 Information of datasets that are used in label compression and recovery experiments and multi-label prediction experiments. In the table, n refers to the number of samples, p refers to the number of features, k refers to the number of labels, K refers to the number of unique label vectors, "Card" refers to the average cardinality of all label vectors, "Density" refers to the average nonzero entry proposition of all label vectors

ID	Datasets	Domain	n	p	k	K	Card	Density
1	Bibtex	Text	7395	1836	159	2856	2.402	0.015
2	Corel5k	Image	5000	499	374	3175	3.522	0.009
3	Mediamill	Video	43907	120	101	6555	4.376	0.043
4	IMDB	Text	120919	1001	28	4503	1.9997	0.0714
5	Enron	Text	1702	1001	53	753	3.378	0.064
6	Genbase	Genomics	662	1186	27	32	1.252	0.046
7	Medical	Text	978	1449	45	94	1.245	0.028
8	Emotions	Music	593	72	6	27	1.869	0.311
9	Scene	Scene	2407	294	6	15	1.074	0.179
10	Slashdot	Text	3782	1079	22	156	1.1809	0.0537
11	Yahoo-Arts	Web	5000	462	26	462	1.6360	0.0629
12	Yahoo-Business	Web	5000	438	30	161	1.5878	0.0529
13	Yahoo-Computers	Web	5000	681	33	253	1.5082	0.0457
14	Yahoo-Education	Web	5000	550	33	308	1.4606	0.0443
15	Yahoo-Entertainment	Web	5000	640	21	232	1.4204	0.0676
16	Yahoo-Health	Web	5000	612	32	257	1.6622	0.0519
17	Yahoo-Recreation	Web	5000	606	22	322	1.4232	0.0647
18	Yahoo-Reference	Web	5000	793	33	217	1.1694	0.0354
19	Yahoo-Science	Web	5000	743	40	398	1.4506	0.0363
20	Yahoo-Social	Web	5000	1047	39	226	1.2834	0.0329
21	Yahoo-Society	Web	5000	636	27	582	1.6920	0.0627

the test samples are predicted as 0, and thus it cannot indicates a successful prediction in this case. Precision and recall should be considered together, because high precision always accompanies low recall when most positive samples are falsely predicted as positive. F1-score and accuracy are less sensitive to the imbalance of label matrix. Therefore, the evaluation of prediction performance should be a integrative consideration of all the five metrics with rough importances according to $F1\text{-}score, Accuracy > Precision, Recall > Hammingloss$.

5.2 Datasets

We evaluate the performance of label compression and recovery, and multi-label prediction of CL on 21 datasets from different domains and of different scales, including Bibtex (Katakis et al. 2008), Corel5k (Duygulu et al. 2002), Mediamill (Snoek et al. 2006), IMDB (Read 2010), Enron (Tsoumakas 2010), Genbase (Diplaris et al. 2005), Medical (Tsoumakas 2010), Emotions (Trohidis et al. 2008), Scene (Boutell et al. 2004), Slashdot (Read 2010) and 11 sub datasets included in Yahoo dataset (Ueda and Saito 2002). These datasets are collected from different practical problems such as text classification, image annotation, scene classification, music categorization, genomics and web page classification. Table 1 shows the number of samples n, number of features p, number of labels k and number of unique labelsets K, the average cardinality of all label vectors *Card*, and the average nonzero entry proportion of all label vectors *Density* of different datasets.

5.3 Label compression and recovery

In this group of experiments, we test the sample balance and label independence on the compressed CL labels, and the recovery accuracy of the corresponding recovery algorithm given in Algorithm 2. We only test the label compression and recovery methods proposed in CL in this group of experiments, and leave the classification and prediction to the second group of experiments. That is because the classification error in CL is caused by two issues, i.e., the label recovery error introduced by compression and the classification error on CL labels. Since the former one is determined by the proposed CL, while the latter one is mainly determined by the binary classification method CL invokes, independent evaluations of different errors are important to the analysis of the effectiveness of CL.

For each of the datasets listed in Table 1, we compress the label matrix to its random projection signs and extract its distilled labelsets by using DL algorithm given in Algorithm 1, and then the recovery algorithm of CL given in Algorithm 2 is invoked to recover the original labels from their CL labels. We measure the degree-of-balance and the χ^2 score of CL labels, and calculate the Hamming loss between the original labels and the recovered ones on different compression ratios. For the two datasets with 6 labels, i.e., Scene and Emotions, the compression ratio is settled from 0 to 3, because the empirical joint distribution estimation will become unstable when the number of random projection signs is too small. For the other datasets, the compression ratio changes from 0 to 1. There are two parameters, i.e., the threshold τ in DL algorithm and the cardinality range in the recovery algorithm. We select $0.01 \leq \tau \leq 0.25$ in all the experiments. In particular, a large τ is recommended for datasets with a large number of unique labelsets, because a small τ often generates clusters without shared label subset in this case. The cardinality range is chosen as the cardinality range of the rows in the original label matrix.

We show the experimental results of label compression and recovery on multiple datasets in Figs. 4 to 6.

In the all the figures, the degree-of-balance of CL labels are kept 0.5 ± 0.1 on different compression ratios. For several datasets with large number of labels, e.g., Bibtex and Corel5k, the degree-of-balance stays very close to the ideal value 0.5. Compared to the degree-of-balance of the original labels, which is shown in the figures as well, CL labels highly improve the sample balance.

In the experiments, the χ^2 score is used to measure the label independence, namely, a larger χ^2 score indicates different labels are more independent. In χ^2 test, when the degree of freedom is 1, a χ^2 score larger than 10.83 associates with a P-value 0.001 and thus indicates the two discrete variables are independent with a very high probability 99.9%. In the figures, most χ^2 scores of the CL labels are much higher than 10.83 and thus the different CL labels are independent. This result is consistent with our theoretical analysis. Compared to the χ^2 score of the original labels, CL labels removes the label dependence, and thus the sequel classification by using conventional binary classification method will be proper and will not downgrade the prediction performance.

The recovery error rates in the figures show that the recoveries are still very accurate with very small Hamming losses when the compression ratio is very small. Thus the computational complexity of CL can be tremendously decreased by using a very low dimensionality, while the learning performance will not be jeopardized. Another observation is that the Hamming loss drops quite fast when the compression ratio is increased. This leads to an efficient compression and precise recovery. Some noises appear on the Hamming loss curve in some figures and makes the curves not monotonically deceasing. That is because the random vectors in A are randomly selected on the hypersphere and thus their distribution on

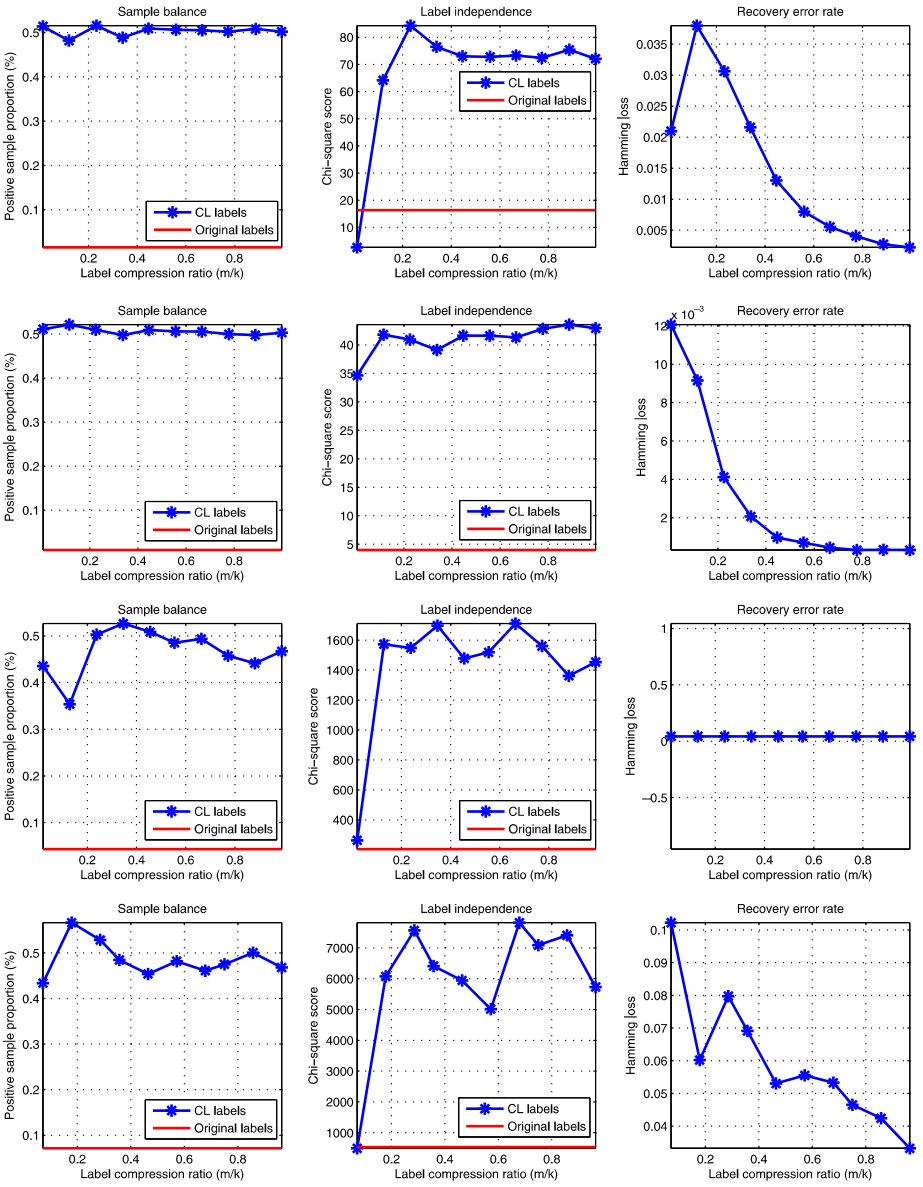

Fig. 4 Sample balance, label independence and recovery error rate on 21 datasets (1). From *top* to *bottom*: Bibtex, Corel5k, Mediamill, IMDB, Enron, Genbase, Medical

the hypershpere is not absolutely even. However, these noises are of small amplitude and thus will not harm the recovery accuracy.

5.4 Multi-label prediction: comparison with BR

In this group of experiments, we compare the multi-label prediction performance and the time cost of CL with BR on 21 datasets with different parameter settings. Since we choose

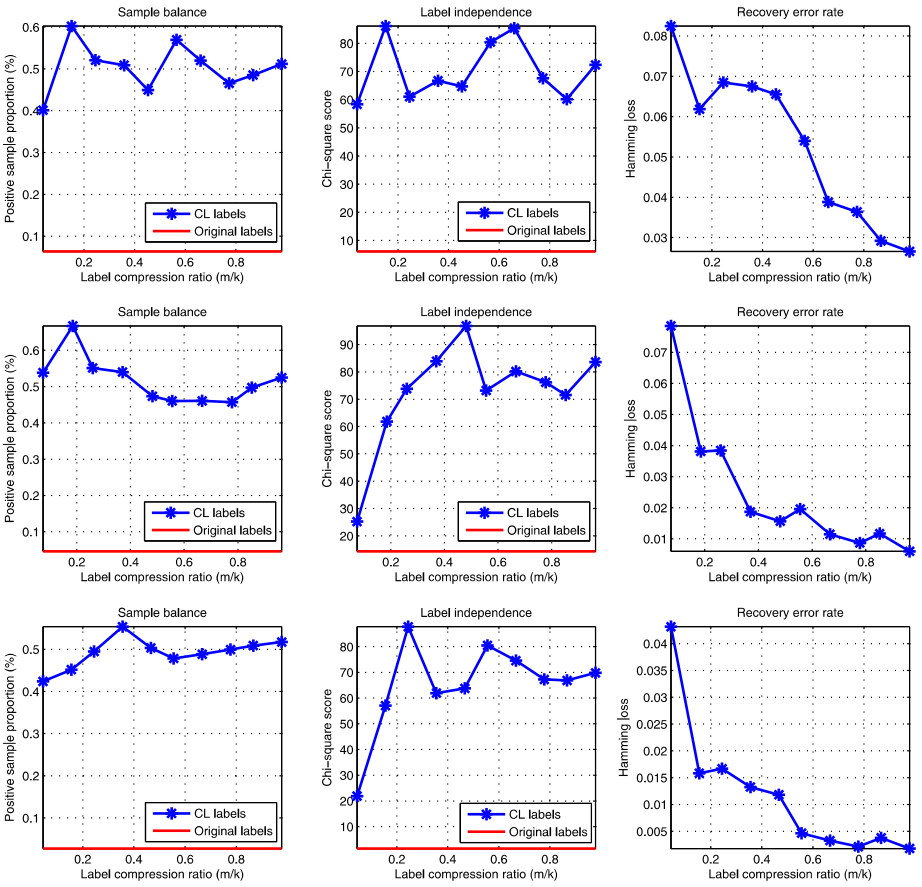

Fig. 4 (*Continued*)

SVM as the single-label learning method in the training stage of CL, BR is compared with CL in the experiments as a baseline method by applying SVM directly on the original labels. This group of experiments aims at verifying the improvement and robustness of prediction performance brought by the training on the CL labels rather than on the original ones. We use linear SVM in the training stages of both BR and CL. This is because linear SVM has only one parameter C, thus the robustness of the performances can be conveniently compared under different parameter settings. The broadly used standard SVM solver "LIBSVM (Chang and Lin 2001)" is applied to all the datasets except Mediamill and IMDB, which are of large scale and thus intractable for LIBSVM. For these two datasets, we alternatively apply NESVM (Zhou et al. 2010), which is a fast gradient SVM solver for large-scale problems.

Table 2 summarizes the information about the training set, the test set and the obtained distilled labelsets of each datasets used in the multi-label prediction experiments. Both training set and test set are randomly selected from the original datasets. The number of distilled labelsets for most datasets is between the number of labels and that of the unique labelsets. When the unique labelsets is of large size, the number of distilled labelsets keeps close to

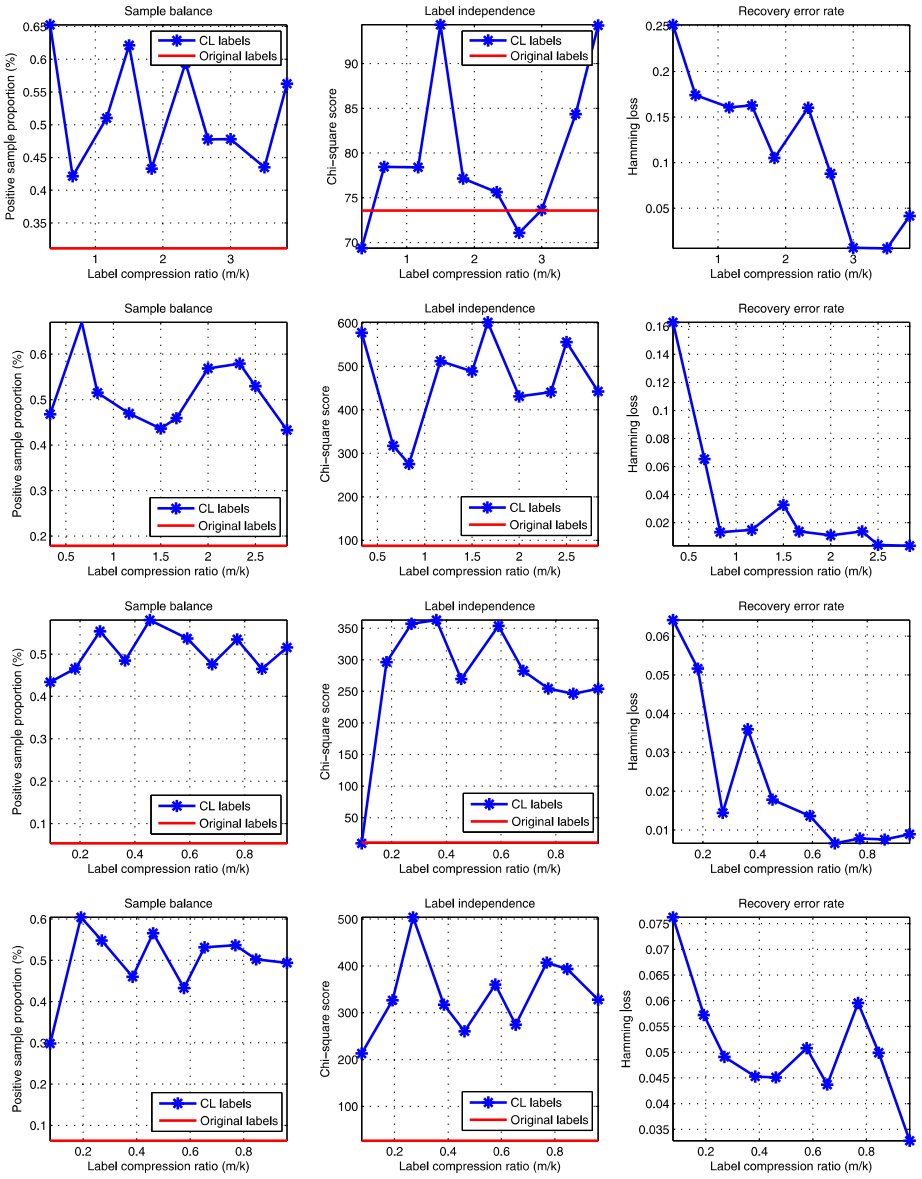

Fig. 5 Sample balance, label independence and recovery error rate on 21 datasets (2). From *top* to *bottom*: Emotions, Scene, Slashdot, Yahoo-Arts, Yahoo-Business, Yahoo-Computers, Yahoo-Education

the number of labels. Thus the computational complexity of recovery algorithm will not be significantly increased when the number of unique labelsets is augmented.

In each experiment, the training algorithm given in Algorithm 3 and the prediction algorithm given in Algorithm 4 are applied to training set and test set, respectively. We test the performance on different C in SVM. The parameter C can be interpreted as the weight of hinge loss in the objective function of SVM. The threshold τ and the cardinality range are

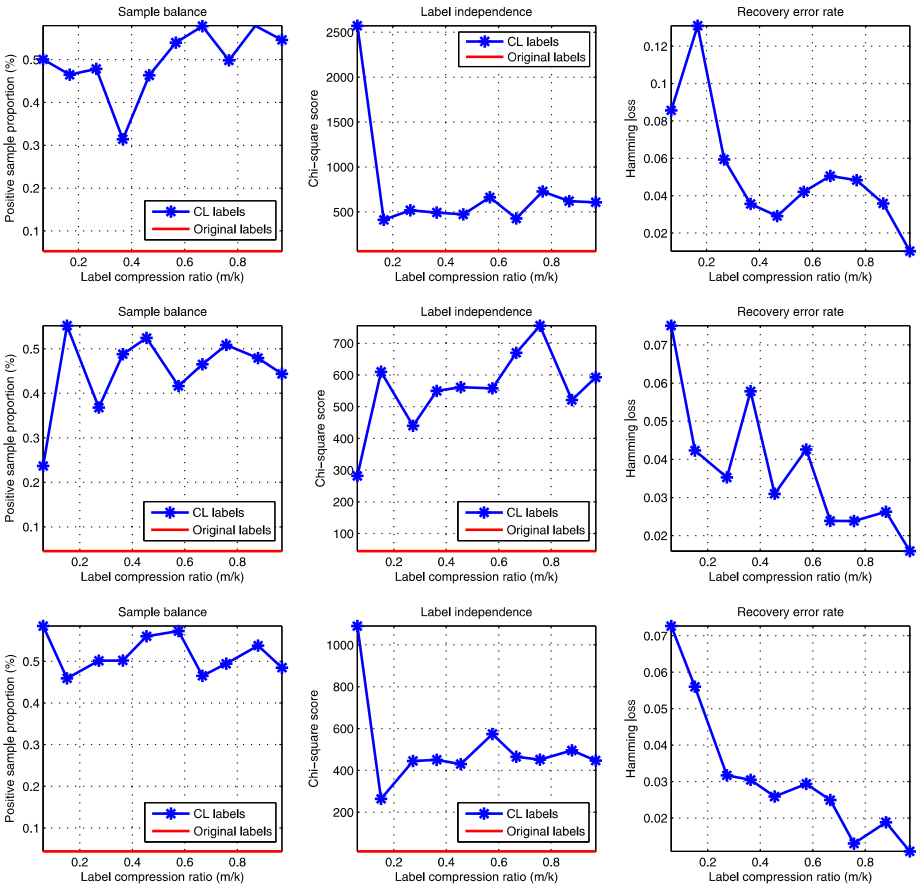

Fig. 5 (*Continued*)

chosen according to the same method in the label compression and recovery experiments. BR is compared with CL on 3-7 different C values between 10^{-3} to 10^{3} in all experiments. For most datasets, 4 C values $[10^{-3}, 10^{-2}, 10^{-1}, 1]$ are used. Since BR fails on some datasets with these C values when it overwhelmingly predicts all the test samples as negative, we choose different C values in this case to guarantee a thorough comparison on sufficient C values.

The prediction performance is evaluated by five metrics, which are Hamming loss, precision, recall, F1 score and accuracy. The good performance is indicated by a small Hamming loss with the large values of the other four metrics. The evaluation of performance should be an integrative consideration of all the five metrics, because one good metric is possibly associated with poor other metrics and a trivial prediction result. For instance, a small hamming loss can associate with other four metrics that are nearly zeros, when most test samples are predicted as negative in all the binary classifications and the number of positive samples is very small. Another example is that a high precision can associate with other four small metrics when the number of positive samples in the predicted one is very small. Compared with precision and recall, F1 score and accuracy are more robust to the above problems.

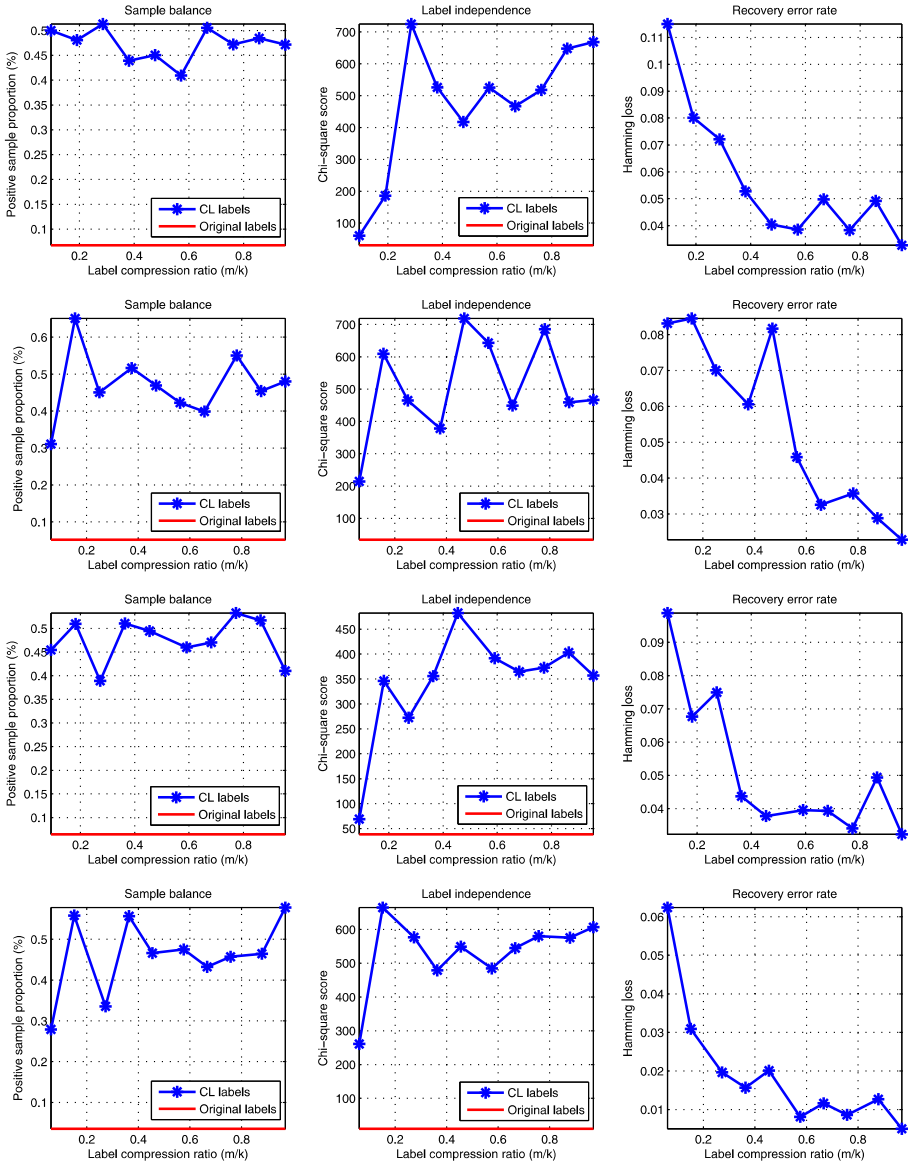

Fig. 6 Sample balance, label independence and recovery error rate on 21 datasets (3). From *top* to *bottom*: Yahoo-Entertainment, Yahoo-Health, Yahoo-Recreation, Yahoo-Reference, Yahoo-Science, Yahoo-Social, Yahoo-Society

We show the multi-label prediction experimental results of CL and BR given different C values on 5 large-scale datasets in Table 3. In this table, we use "–" as the mark of prediction failure, namely, predicting all the samples into one class in each binary classification. The "Time" column in the tables is the sum of training and test CPU seconds. Thus the CPU seconds for label compression and recovery in CL is included in "Time". The best performance of BR and CL are shadowed with different colors in the table. For each dataset, we

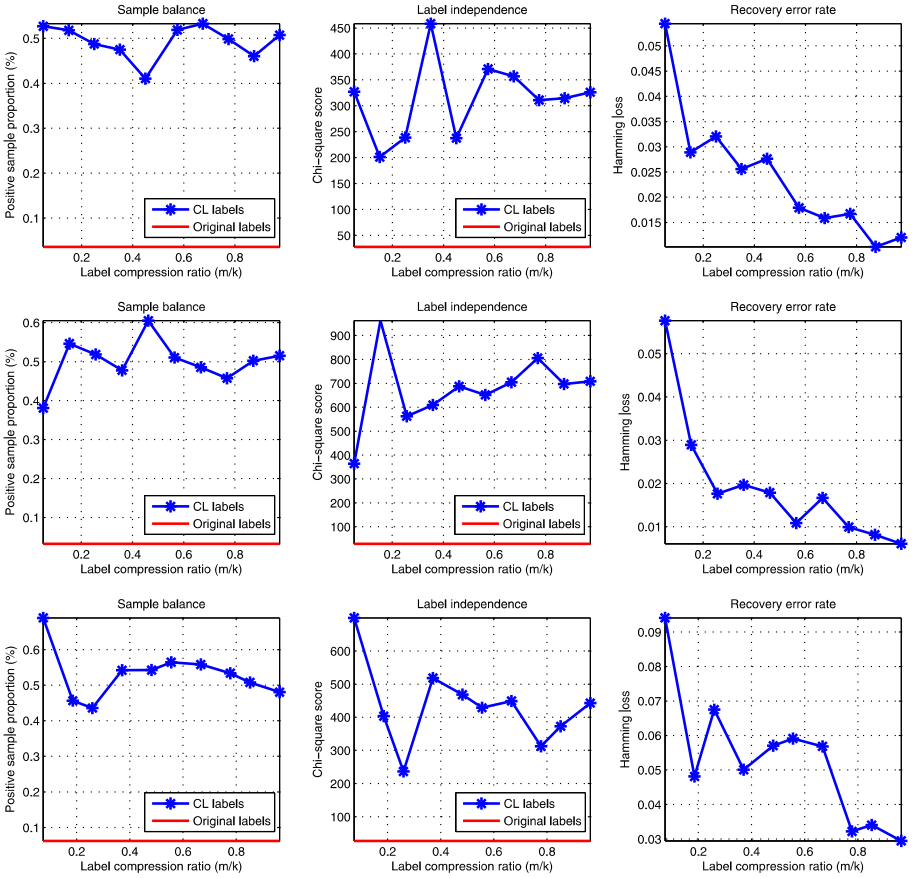

Fig. 6 (*Continued*)

choose the number of CL labels much less than the number of the original ones except in datasets Scene and Emotions, where the numbers of labels are too small (6) and thus random projection signs less than it are not stable for estimating the empirical joint distribution.

In most experiments, CL significantly outperforms BR on different metrics and different values of C. In particular, CL has overwhelming privilege on large-scale datasets with large number of labels (which is one of the most difficult cases in multi-label learning), e.g., Bibtex, Corel5k, Enron and Medical. CL also has appealing prediction performance on small datasets whose density is small, e.g., Genbase, Slashdot and most sub datasets of Yahoo. Another interesting phenomenon is that the dimension increasing of label matrix on Scene dataset generates a satisfying prediction. This indicates that CL can also be used to improve multi-label prediction on dataset with small number of labels by increasing the dimensionality of CL labels. There exist rare cases when BR has better performance, e.g., on Yahoo-Business with $C = 10^3$ and $C = 10^2$, but CL and BR arrive the same metrics when $C = 10^1$ and $C = 1$.

CL also brings a tremendous reduction of the time cost. According to the CPU seconds and the number of labels shown in the tables, CL can compress the original label matrix to a low dimensionality, which saves a great amount of computation in the training stage.

Table 2 Training set size, test set size and the obtained distilled labelsets size of each datasets in the multi-label prediction experiments. In order to compare the number of distilled labelsets d with the number of labels k and the number of unique labelsets K, we list k and K of each datasets in the table as well

ID	Datasets	Training	Test	DL	k	K
1	Bibtex	4880	2515	253	159	2856
2	Corel5k	4500	500	418	374	3175
3	Mediamill	30993	12914	161	101	6555
4	IMDB	62050	58869	61	28	4503
5	Enron	1123	579	73	53	753
6	Genbase	463	199	27	27	32
7	Medical	333	645	46	45	94
8	Emotions	391	202	14	6	27
9	Scene	1211	1196	6	6	15
10	Slashdot	2338	1444	20	22	156
11	Yahoo-Arts	2000	3000	37	26	462
12	Yahoo-Business	2000	3000	35	30	161
13	Yahoo-Computers	2000	3000	39	33	253
14	Yahoo-Education	2000	3000	35	33	308
15	Yahoo-Entertainment	2000	3000	26	21	232
16	Yahoo-Health	2000	3000	35	32	257
17	Yahoo-Recreation	2000	3000	31	22	322
18	Yahoo-Reference	2000	3000	38	33	217
19	Yahoo-Science	2000	3000	47	40	398
20	Yahoo-Social	2000	3000	47	39	226
21	Yahoo-Society	2000	3000	39	27	582

Therefore, CL is capable to reduce the problem size of multi-label learning and makes the large-scale problems computationally tractable.

Since the unique difference between CL and BR is that CL invokes linear SVM on the compressed labels rather than on the original ones as BR does, this group of experimental results shows that single-label classification methods can be conveniently extended to solve multi-label learning problems with improved performance and reduced time cost by using the label compression and recovery scheme proposed in CL. The leverages in both effectiveness and efficiency are attributed to the removal of the three main problems of applying single-label learning method to multi-label learning tasks. In particular, the performance improvement benefits from the sample balance and exploration of label correlation in CL, while the time cost is decreased by label compression. In addition, the benefits and the strength of conventional single-label learning methods are well inherited in the scheme of CL. Therefore, CL proposes a general extension of single-label learning method to multi-label scenarios rather than merely a specific algorithm for multi-label learning.

CL is robust to the change of parameters in the training stage. In the experimental results, CL can generate appealing prediction result when BR fails on some C values. Moreover, its performance changes a little when C decreases from 10^3 to 10^{-3}, e.g., on the Scene dataset. We explain the reasons for the improvement of robustness on two aspects:

1. SVM is more robust to the parameter C on more balanced data. In imbalanced data, the insufficient number of samples in the minor class are more likely to be within the mar-

Table 3 Multi-label performances and time costs of BR and CL on 10 datasets with different C parameters (1). HL-Hamming Loss, Prec-Precision, Rec-Recall, Acc-Accuracy, Time-CPU seconds, labels-Number of Labels in training stage. "–" denotes the failed experiment that incorrectly predicts all the test samples as negative. The best performances of BR and CL are highlighted with different colors

	C		HL	Prec	Rec	F1	Acc	Time	Labels
Bibtex	10^3	BR	0.1261	0.0158	0.1227	0.0258	0.0139	4630	159
		CL	0.1070	0.1609	0.5876	0.2049	0.1403	1580	80
	10^2	BR	0.0638	0.0158	0.0422	0.0197	0.0112	5310	159
		CL	0.1032	0.1912	0.6009	0.2315	0.1651	1450	80
	10^1	BR	0.2242	0.0118	0.1417	0.0211	0.0110	5890	159
		CL	0.1075	0.2210	0.5658	0.2445	0.1839	1640	80
	1	BR	–	–	–	–	–	–	159
		CL	0.1509	0.1453	0.4493	0.1618	0.1204	1810	80
Corel5k	10^3	BR	0.0871	0.0252	0.2035	0.0439	0.0231	2240	374
		CL	0.0157	0.2236	0.2365	0.2201	0.1473	911.4	160
	10^2	BR	0.0869	0.0139	0.1027	0.0232	0.0122	1870	374
		CL	0.0153	0.2418	0.2595	0.2392	0.1585	526.5	160
	10^1	BR	0.0311	0.0074	0.0167	0.0102	0.0057	1270	274
		CL	0.0164	0.2051	0.2473	0.2181	0.1397	476.1	160
	1	BR	0.2101	0.0093	0.2011	0.0178	0.0091	1050	374
		CL	0.0159	0.1957	0.2245	0.2058	0.1295	450.9	160

Table 3 (*Continued*)

	C		HL	Prec	Rec	F1	Acc	Time	Labels
Mediamill	10^3	BR	0.036	0.6913	0.3545	0.4315	0.3308	203.3	101
		CL	0.040	0.5680	0.4630	0.4601	0.3436	77.94	40
	10^2	BR	0.036	0.6913	0.3545	0.4315	0.3308	202.4	101
		CL	0.043	0.5141	0.4635	0.4412	0.3210	72.21	40
	10^1	BR	0.036	0.6913	0.3545	0.4315	0.3308	205.1	101
		CL	0.039	0.5751	0.4629	0.4659	0.3467	76.70	40
	1	BR	0.036	0.6913	0.3545	0.4315	0.3308	204.8	101
		CL	0.063	0.4076	0.5637	0.4191	0.2610	76.03	40
IMDB	10^3	BR	0.0689	0.6338	0.0024	0.0061	0.0024	5352	28
		CL	0.1321	0.2825	0.4567	0.3035	0.2186	3628	20
	10^2	BR	0.0689	0.6338	0.0024	0.0060	0.0024	5318	28
		CL	0.1492	0.2621	0.5178	0.3080	0.2187	3273	20
	10^1	BR	–	–	–	–	–	3529	28
		CL	0.1737	0.2408	0.5048	0.2743	0.1944	1434	20
	1	BR	–	–	–	–	–	2700	28
		CL	0.1971	0.1646	0.5431	0.2254	0.1463	898.2	20

Table 3 (*Continued*)

	C		HL	Prec	Rec	F1	Acc	Time	Labels
Enron	10^3	BR	0.2283	0.1482	0.5184	0.2183	0.1280	122.8	53
		CL	0.0971	0.3940	0.5080	0.4053	0.2887	36.59	36
	10^2	BR	0.2031	0.1633	0.4360	0.2229	0.1347	168.6	53
		CL	0.0854	0.4549	0.5542	0.4580	0.3370	31.00	36
	10^1	BR	0.0915	0.1307	0.2762	0.2552	0.1718	147.2	53
		CL	0.0779	0.5027	0.5926	0.5076	0.3905	35.60	36
	1	BR	0.0636	0.4957	0.2733	0.3429	0.2324	1050	53
		CL	0.0548	0.3949	0.6019	0.4267	0.2929	37.56	36
	10^{-1}	BR	0.0624	0.5138	0.2851	0.3573	0.2415	77.10	53
		CL	0.1336	0.2708	0.6726	0.3693	0.2398	29.77	36
	10^{-2}	BR	0.1372	0.1742	0.2888	0.2115	0.1250	56.74	53
		CL	0.1020	0.2957	0.5015	0.3510	0.2219	26.11	36
	10^{-3}	BR	0.0624	0.5138	0.2851	0.3573	0.2410	39.93	53
		CL	0.1165	0.2990	0.6482	0.3911	0.2588	15.58	36

gin when C decreases, i.e., when the margin becomes larger. This leads to less support vectors for the minor class. Thus the prediction performance of BR drops dramatically with the decreasing of C. CL trains SVM classifiers on compressed labels with improved sample balance, and thus the prediction performance is more robust to a wide range of C.

2. In CL, although the number of labels are reduced due to the label compression, the number of labels that each sample belongs to is increased, because the label matrix is much more dense and balanced. Thus the information of the original labels are disseminated rather than condensed in the CL labels. In the prediction stage, each original label is predicted according to the statistics of all the CL labels. This scheme makes the prediction of the original label robust to the failures in predicting one or two CL labels.

Therefore, CL improves the robustness of the multi-label learning via random projection signs of the labels.

5.5 Multi-label prediction: comparison with other multi-label learning methods

In this group of experiments, we compare CL with 3 popular multi-label learning methods, i.e., ML-knn (Zhang and Zhou 2007), MDDM (Zhang and Zhou 2008) and multi-label prediction based CS (ML-CS) (Hsu et al. 2009). ML-knn is an extension of knn. It obtains the label prior distribution from the k nearest neighbors and maximizes a posterior to the label prediction. ML-knn aims at solving the imbalance problem of the k nearest neighbors' labels. MDDM tackles the "curse of dimensionality" in multi-label data and formulates the problem as a discriminative dimension reduction (Tao et al. 2009). It maximizes the dependence between feature space and label space via maximizing the empirical estimate of Hilbert-Schmidt Independence Criterion. Then ML-knn can be applied to the obtained low-dimensional subspace. MDDM aims at decreasing the time complexity of the sequential ML-knn.

ML-CS compresses the original 0–1 label matrix Y via random projection $Y' = YA$ to a real valued matrix Y', and then builds a new linear regression model $Y' = XW'$. In its prediction, the original label vector y is recovered from y' via compressed sensing or model selection algorithms such as least angle regression (LARS) (Efron et al. 2002). ML-CS is related to CL because it builds models on the low-dimensional embedding of the original label. Their main difference is that CL compresses the original 0–1 label matrix to another 0–1 label matrix and builds binary classification models on it rather than regression models in the training stage.

In the experiments, we set the number of neighbors in ML-knn as 20 and the dimensions of the subspace obtained in MDDM as 40% of the dimensions of the original data. In ML-CS, the low-dimensional embedding of the label matrix has the same dimension as the compressed label matrix in CL. We use LARS as the recovery algorithm in its prediction stage. The sparse level is determined by selecting the solution with the minimum least square measurement error in the same cardinality range as CL. This is similar to the last step in the recovery algorithm of CL in Algorithm 2. The performance of CL is collected from the first group of experiments.

We show the experimental results in Tables 4 and 5.

In most experiments, CL outperforms the other 3 multi-label learning methods on most performance metrics. Since ML-knn is more robust to sample imbalance than BR based on SVM, and ML-CS eliminates the sample imbalance via random projections, sample balance problem are considered in these 3 methods. Moreover, ML-knn explores the label correlation by its instance-based learning scheme. Thus the 3 methods have better prediction performance than BR in the first group of experiments. However, ML-knn and MDDM are vulnerable to dataset with ultra sample imbalance, e.g., Bibtex, Corel5k and Mediamill, where

Table 4 Prediction performances and time costs of ML-knn, MDDM, ML-CS and CL on 10 datasets. "–" denotes the failed experiment whose time cost exceeds 10^5 seconds

	Methods	HL	Prec	Rec	F1	Acc	Time	Labels
Bibtex	ML-knn	0.0140	0.5511	0.0397	0.2034	0.0364	42697	159
	MDDM	0.0140	0.5334	0.0534	0.2138	0.0492	970.7	159
	ML-CS	0.1645	0.0798	0.6499	0.1026	0.0760	1901.5	80
	CL	0.1075	0.2210	0.5658	0.2445	0.1839	1640	80
Corel5k	ML-knn	0.0093	0.6197	0.0103	0.0321	0.0098	2106	374
	MDDM	0.0093	0.6166	0.0161	0.0755	0.0146	458	374
	ML-CS	0.0857	0.0789	0.5108	0.1012	0.0741	612.6	160
	CL	0.0153	0.2418	0.2595	0.2392	0.1585	526.5	160
Mediamill	ML-knn	0.0314	0.4132	0.0632	0.5377	0.0534	5713	101
	MDDM	0.0319	0.3679	0.0517	0.5278	0.0439	48237	101
	ML-CS	0.1116	0.2503	0.7033	0.3402	0.2206	99.48	40
	CL	0.0390	0.5751	0.4629	0.4659	0.3467	76.70	40
IMDB	ML-knn	–	–	–	–	–	$> 10^5$	28
	MDDM	–	–	–	–	–	$> 10^5$	28
	ML-CS	0.1890	0.1654	0.4228	0.2286	0.1406	3608	20
	CL	0.1492	0.2621	0.5178	0.3080	0.2187	3273	20
Enron	ML-knn	0.0518	0.5083	0.0665	0.4632	0.0532	527	53
	MDDM	0.0505	0.5000	0.0883	0.4966	0.0715	29	53
	ML-CS	0.3857	0.1315	0.6732	0.1747	0.1189	40.36	36
	CL	0.0779	0.5027	0.5926	0.5076	0.3905	35.60	36
Genbase	ML-knn	0.0065	1.0000	0.5071	0.9231	0.5071	9.38	32
	MDDM	0.0063	0.9881	0.5170	0.9258	0.5133	6.09	32
	ML-CS	0.0143	0.9140	0.9949	0.8627	0.9104	0.327	20
	CL	0.0101	0.9605	0.9728	0.9568	0.9441	0.321	20
Medical	ML-knn	0.0204	0.7554	0.0743	0.4879	0.0657	22.8	45
	MDDM	0.0249	0.7395	0.0323	0.3058	0.0258	32.3	45
	ML-CS	0.3306	0.0896	0.9356	0.1334	0.0893	3.14	25
	CL	0.0273	0.5904	0.8147	0.6565	0.5694	1.43	25
Emotions	ML-knn	0.2855	0.6852	0.2819	0.4155	0.2237	0.66	6
	MDDM	0.2962	0.5437	0.2885	0.4163	0.2239	0.66	6
	ML-CS	0.3960	0.4339	0.6113	0.5171	0.3927	0.68	15
	CL	0.3201	0.5208	0.6007	0.5353	0.4373	0.51	15
Scene	ML-knn	0.1016	0.7788	0.6257	0.6881	0.5373	14.3	6
	MDDM	0.1044	0.7568	0.6393	0.6870	0.5338	7.59	6
	ML-CS	0.4123	0.2672	0.6768	0.3734	0.2635	20.42	15
	CL	0.0780	0.6080	0.8023	0.6627	0.5887	16.50	15
Slashdot	ML-knn	0.0481	0.7104	0.1016	0.3067	0.0800	708	22
	MDDM	0.0518	0.3973	0.0129	0.0452	0.0118	114	22
	ML-CS	0.3866	0.0881	0.6774	0.1530	0.0866	31.86	12
	CL	0.0833	0.4269	0.6514	0.4913	0.4122	27.68	12

Table 5 Prediction performances and time costs of ML-knn, MDDM, ML-CS and CL on 11 sub datasets from Yahoo dataset

	Methods	HL	Prec	Rec	F1	Acc	Time	Labels
Arts	ML-knn	0.0587	0.6257	0.0708	0.2477	0.0639	77.6	26
	MDDM	0.0595	0.6783	0.0614	0.2123	0.0562	37.4	26
	ML-CS	0.4538	0.1004	0.7994	0.1753	0.0975	35.3	22
	CL	0.1836	0.3032	0.6452	0.3583	0.2693	29.7	22
Business	ML-knn	0.0266	0.6789	0.0920	0.7042	0.0822	93.2	30
	MDDM	0.0279	0.6609	0.0733	0.6878	0.0661	42.7	30
	ML-CS	0.1089	0.3975	0.8186	0.4246	0.3663	30.3	25
	CL	0.0287	0.8633	0.6797	0.7332	0.6797	21.9	25
Computers	ML-knn	0.0408	0.6959	0.0344	0.3328	0.0303	124	33
	MDDM	0.0414	0.5669	0.0497	0.4075	0.0403	50	33
	ML-CS	0.1329	0.3149	0.6432	0.2914	0.2817	54.0	25
	CL	0.0736	0.4846	0.6850	0.5196	0.4334	51.6	25
Education	ML-knn	0.0389	0.5883	0.0618	0.3159	0.0560	99.8	33
	MDDM	0.0398	0.5914	0.0502	0.2655	0.0468	45.2	33
	ML-CS	0.1382	0.2406	0.6218	0.2755	0.2188	40.9	25
	CL	0.1079	0.3123	0.6383	0.3749	0.2844	36.0	25
Entertain	ML-knn	0.0575	0.6662	0.1126	0.3328	0.1067	108	21
	MDDM	0.0585	0.6485	0.1093	0.3005	0.1040	43.8	21
	ML-CS	0.2202	0.2192	0.5965	0.2550	0.2027	28.8	16
	CL	0.1433	0.4028	0.6573	0.4500	0.3697	28.2	16
Health	ML-knn	0.0362	0.7316	0.1618	0.5657	0.1471	101	32
	MDDM	0.0391	0.7040	0.1448	0.5138	0.1310	46.6	32
	ML-CS	0.1092	0.3739	0.7319	0.3900	0.3384	23.3	25
	CL	0.0299	0.6140	0.7062	0.6039	0.5636	21.1	25
Recreation	ML-knn	0.0595	0.7016	0.0862	0.2310	0.0804	112	22
	MDDM	0.0612	0.6655	0.0725	0.1801	0.0671	41.9	22
	ML-CS	0.2264	0.1794	0.5383	0.2228	0.1653	24.6	12
	CL	0.2210	0.3125	0.6966	0.3721	0.2864	23.7	12
Reference	ML-knn	0.0274	0.6663	0.0699	0.4696	0.0638	136	33
	MDDM	0.0290	0.6620	0.0676	0.4325	0.0598	51.6	33
	ML-CS	0.1404	0.2610	0.6823	0.2484	0.2480	24.1	25
	CL	0.0161	0.4892	0.6582	0.5252	0.4618	36.2	25
Science	ML-knn	0.0330	0.5999	0.0479	0.2056	0.0446	139	40
	MDDM	0.0336	0.6603	0.0456	0.1986	0.0424	53	40
	ML-CS	0.1851	0.1496	0.6130	0.1792	0.1405	45.8	30
	CL	0.0957	0.2725	0.5939	0.3381	0.2521	59.1	30

Table 5 (*Continued*)

	Methods	HL	Prec	Rec	F1	Acc	Time	Labels
Social	ML-knn	0.0219	0.7759	0.0718	0.5537	0.0675	175	39
	MDDM	0.0238	0.6751	0.0577	0.5003	0.0535	70.8	39
	ML-CS	0.1443	0.2215	0.7271	0.2386	0.2123	65.2	29
	CL	0.0597	0.5642	0.7544	0.5913	0.5288	59.0	29
Society	ML-knn	0.0547	0.6258	0.0574	0.3422	0.0499	111	27
	MDDM	0.0542	0.6077	0.0530	0.3136	0.0488	44.1	27
	ML-CS	0.2130	0.2214	0.5983	0.2351	0.1975	49.9	19
	CL	0.1429	0.3785	0.5924	0.4044	0.3191	43.2	19

they have high precision and F1 score but near 0 recall and accuracy. ML-knn and MDDM have similar performance on most datasets, but MDDM exceeds ML-knn on datasets of high data dimensionality such as Bibtex and Enron. This indicates the advantages of MDDM on high dimensional data when ML-knn is applied as the classification model. ML-CS performs better than ML-knn and MDDM because of the independence among the dimensions of the label embedding brought by random projection. The main reason that ML-CS does not outperform CL is that ML-CS transforms the original discrete classification model to a continuous linear regression model, which small estimation error may induce large error in the label recovery. Although both the estimation error of linear regression model and the recovery error of the compressed sensing algorithm can be theoretically bounded, it is not clear how the linear regression estimation error as an input of the recovery influences the final classification error. CL remains training discrete classification model (of smaller size than the original one) on CL labels and thus does not meet this problem. Another advantage of CL comparing with ML-CS is that the label correlations are explicitly and directly explored via LDM, and thus the prediction is improved.

The time cost of CL is less than those of ML-knn and MDDM in all the experiments, especially on the large-scale datasets such as Bibtex, Corel5k and Mediamill. ML-knn and MDDM fail on IMDB because we fail to train them in 10^5 CPU seconds. This is due to the expensive time complexity $\mathcal{O}(n^2 p)$ of the pairwise distance calculation in ML-knn and MDDM. MDDM is more efficient than ML-knn with a decreased p. The time costs of CL and ML-CS are close to each other, and CL performs slightly faster than ML-CS on most datasets. However, it is worthy to note that CL and ML-CS have different training time and prediction time. In particular, the training of SVM in CL is slower than the training of linear regression in ML-CS. But the prediction time of CL is much less than that of ML-CS, because the recovery algorithm in CL is based on a simple statistical test, while the recovery algorithm in ML-CS invokes time consuming ℓ_1 minimization. In addition, CL can be substantially faster than ML-CS if we replace LIBSVM with other efficient SVM solvers such as NESVM.

5.6 Multi-label prediction: comparison with 2 SVM algorithms dealing with imbalanced data

In this group of experiments, we compare CL with 2 representative SVM extensions dealing with the sample imbalance problem, i.e., SVM-SMOTE (Chawla et al. 2002) and SVM-WEIGHT (Osuna et al. 1997). The goal of these experiments is to (1) study whether the

Table 6 Prediction performances and time costs of SVM-SMOTE, SVM-WEIGHT and CL on 5 datasets

	Methods	HL	Prec	Rec	F1	Acc	Time	Labels
Medical	SMOTE	0.9679	0.0129	0.3964	0.0253	0.0128	2.435	45
	WEIGHT	0.3413	0.0388	0.3078	0.0548	0.0377	2.708	45
	CL	0.0273	0.5904	0.8147	0.6565	0.5694	1.433	25
Emotions	SMOTE	0.2962	0.4600	0.57178	0.4562	0.3982	1.625	6
	WEIGHT	0.2962	0.4547	0.5379	0.4426	0.3805	0.253	6
	CL	0.3201	0.5208	0.6007	0.5353	0.4373	0.517	15
Scene	SMOTE	0.1128	0.5301	0.6182	0.5631	0.5261	23.42	6
	WEIGHT	0.1128	0.5318	0.6207	0.4938	0.5269	9.244	6
	CL	0.0780	0.6080	0.8023	0.6627	0.5887	16.50	15
Genbase	SMOTE	0.9885	0.0086	0.1264	0.0164	0.0082	2.098	32
	WEIGHT	0.5736	0.0497	0.7089	0.0945	0.0488	0.964	32
	CL	0.0101	0.9605	0.9728	0.9568	0.9441	0.321	20
Slashdot	SMOTE	0.9311	0.0270	0.4361	0.0516	0.0268	43.65	22
	WEIGHT	0.3102	0.0797	0.4310	0.1366	0.0778	121.5	22
	CL	0.0833	0.4269	0.6514	0.4913	0.4122	27.68	12
Arts	SMOTE	0.1200	0.3172	0.5360	0.3421	0.2696	115.4	26
	WEIGHT	0.1195	0.3143	0.5300	0.3403	0.2674	54.54	26
	CL	0.1836	0.3032	0.6452	0.3583	0.2693	29.72	22
Business	SMOTE	0.0448	0.6763	0.8006	0.6462	0.5958	86.36	30
	WEIGHT	0.0458	0.6577	0.7171	0.6179	0.5535	38.99	30
	CL	0.0287	0.8633	0.6797	0.7332	0.6797	21.98	25
Computers	SMOTE	0.0724	0.4435	0.5849	0.4161	0.3765	141.1	33
	WEIGHT	0.0709	0.4505	0.5722	0.4154	0.3790	79.93	33
	CL	0.0736	0.4846	0.6850	0.5196	0.4334	51.69	25
Education	SMOTE	0.0678	0.3341	0.5516	0.3529	0.2279	111.9	33
	WEIGHT	0.0680	0.3327	0.5548	0.3531	0.2290	53.89	33
	CL	0.1079	0.3123	0.6383	0.3749	0.2844	36.02	25
Entertain	SMOTE	0.0920	0.4587	0.5948	0.4437	0.4001	121.4	21
	WEIGHT	0.0920	0.4566	0.5882	0.4415	0.3961	62.37	21
	CL	0.1433	0.4028	0.6573	0.4500	0.3697	28.22	16
Health	SMOTE	0.0652	0.5637	0.6616	0.5417	0.4709	102.8	32
	WEIGHT	0.0650	0.5605	0.6567	0.5409	0.4692	48.46	32
	CL	0.0299	0.6140	0.7062	0.6039	0.5636	21.18	25
Recreation	SMOTE	0.109	0.3062	0.5339	0.3116	0.2773	121.5	22
	WEIGHT	0.1081	0.3036	0.5295	0.3208	0.2843	59.22	22
	CL	0.2210	0.3125	0.6966	0.3721	0.2864	23.71	12

Table 6 (*Continued*)

	Methods	HL	Prec	Rec	F1	Acc	Time	Labels
Reference	SMOTE	0.0413	0.5096	0.5366	0.4595	0.4468	123.4	33
	WEIGHT	0.0411	0.5091	0.5330	0.4583	0.4452	69.04	33
	CL	0.0161	0.4892	0.6582	0.5252	0.4618	36.29	25
Science	SMOTE	0.0563	0.3584	0.4780	0.3604	0.3023	186.5	40
	WEIGHT	0.0556	0.3622	0.4669	0.3579	0.3011	97.09	40
	CL	0.0957	0.2725	0.5939	0.3381	0.2521	59.10	30
Social	SMOTE	0.0293	0.6698	0.6374	0.5739	0.5566	163.8	39
	WEIGHT	0.0292	0.6689	0.6356	0.5732	0.5532	87.57	39
	CL	0.0597	0.5642	0.7544	0.5913	0.5288	59.04	29
Society	SMOTE	0.1100	0.3272	0.5022	0.3240	0.2532	173.5	27
	WEIGHT	0.1088	0.3306	0.5038	0.3263	0.2567	92.26	27
	CL	0.1429	0.3785	0.5924	0.4044	0.3191	43.211	19

sample balance obtained via random projection signs in CL works well as the other methods which imposes over-sampling or larger weight to the minor class; (2) evaluate the performance improvement caused by exploring label correlation in CL isolated from that caused by the sample balance. In the experiments of SVM-SMOTE and SVM-WEIGHT, we replace the ordinary SVM in BR with these two modified versions, respectively. SVM-SMOTE invokes SMOTE algorithm to generate synthetic but pseudo samples for minor (positive) class and then train SVM on an dataset with over-sampled positive samples and the original negative ones. We let the positive sample has the same amount of the negative ones after over-sampling. SVM-WEIGHT assigns a larger weight penalty to the major class, which leads to a larger margin for the major class than that for the minor class. We set the weights for the negative samples and the positive ones in proportion to the ratio of their amounts. We show the experimental results in Table 6.

Table 6 shows that CL outperforms SVM-SMOTE and SVM-WEIGHT on both effectiveness and efficiency. Compared with the performance of BR using the ordinary SVM in the first group of experiments, the 2 SVM algorithms are designed to alleviate the harm brought by sample imbalance to SVM, and thus outperform BR. The phenomenon of "high precision low recall" is rare in this group of experiments (though still exists on Genbase and Slashdot), which indicates that the sample balance is helpful for improving multi-label prediction. Different from over-sampling and weighting in SVM-SMOTE and SVM-WEIGHT, CL eliminates the sample balance via random projection signs. The experimental results show this method works well as the other two. Compared with SVM-SMOTE and SVM-WEIGHT, the improvement of prediction performance in CL is due to the application of DLs that preserve label correlation. Therefore, the sample balance and label correlation indeed help to improve the prediction in CL.

SVM-SMOTE adopts over-sampling technique and thus increases the time complexity of the original SVM in BR by augmenting the training samples. SVM-WEIGHT has similar time complexity of the original SVM in BR, but the changing of weight in major (negative) samples may influence the convergence of the SVM training, and thus the time cost will be slightly different. CL decreases the time complexity of BR by reducing the number of SVM

models that need to be trained, and thus has the smallest time cost among the 3 methods in this group of experiments.

5.7 Compression-performance trade-off

In this group of experiments, we study the trade-off between the recovery error due to label compression and the improvement brought by sample balance for prediction performance in CL. In particular, we evaluate the time cost and the 5 performance metrics of CL without using DL on 10 different label compression ratios between 0 and 1. The C parameter for each dataset is selected as the C with the best performance on the same dataset in the first group of experiments. We use NESVM (Zhou et al. 2010) as the SVM solver. In order to isolate the improvement caused by sample balance, we eliminate the influence of label correlation via replacing DLs in CL with the dictionary D composed of unit vectors, i.e.,

$$D = \left\{e^i\right\}_{i=1,\ldots,k}, \quad e^i = [0,\ldots,0,1,0,\ldots,0], e^i_i = 1. \tag{89}$$

In order to ensure that the least label compression ratio can generate more than 2 CL labels, we choose the 5 datasets with the largest number of labels k which is larger than 40 among the 21 datasets and show their trade-off curves in Fig. 7. There are some noises that make the curves not strictly monotonic. These noises are generated by the randomness of CL labels. However, these noises slightly affect the trends of the trade-off curves.

Among the 6 trade-off curves, the time cost linearly grows with the increasing of the label compression ratio. This can be explained by that the increasing of SVM models needs to be trained in CL with the increasing compression ratio. The hamming loss does not affect the trend because it is unstable when the sample imbalance is severe. In this case, a failed prediction that assigns all the test samples to negative classes can produce an extremely small hamming loss. The other 4 metrics, i.e., precision, recall, F1 score and accuracy, rapidly increase with the increasing of the compression ratio. Moreover, they quickly reach satisfactory performances when the compression ratio is fairly small (less than 0.4) on all the 5 datasets. These results indicate that the prediction performance of CL stably and rapidly improves with the increasing of CL labels. Furthermore, CL reaches satisfactory sample balance when the compression ratio is small. Thus CL finds the equilibrium with both competitive efficiency and effectiveness.

Another interesting property shown on the trade-off curves is the improvement brought by exploring the label correlation in CL. For example, the best performance of CL on Bibtex in Fig. 7 with compression ratio 1 and without using of DLs is much less than the performance with compression ratio 0.5 shown in Table 4 where DLs is applied. This difference suggests that LDM preserves label correlations in improving multi-label prediction performance.

6 Conclusion

In this paper, we have proposed a label transformation method "compressed labeling (CL)" for multi-label learning. In the training stage of CL, the original label matrix is compressed to the sign matrix of its low-dimensional random projections on a standard Gaussian ensemble. Existing binary classification is then directly applicable to the labels in the new label matrix. In the prediction stage of CL, the CL label vector of a given sample is initially estimated by using the classifiers obtained in the training stage. Afterward, a fast recovery algorithm is used to reconstruct the original label vector from the CL label vector.

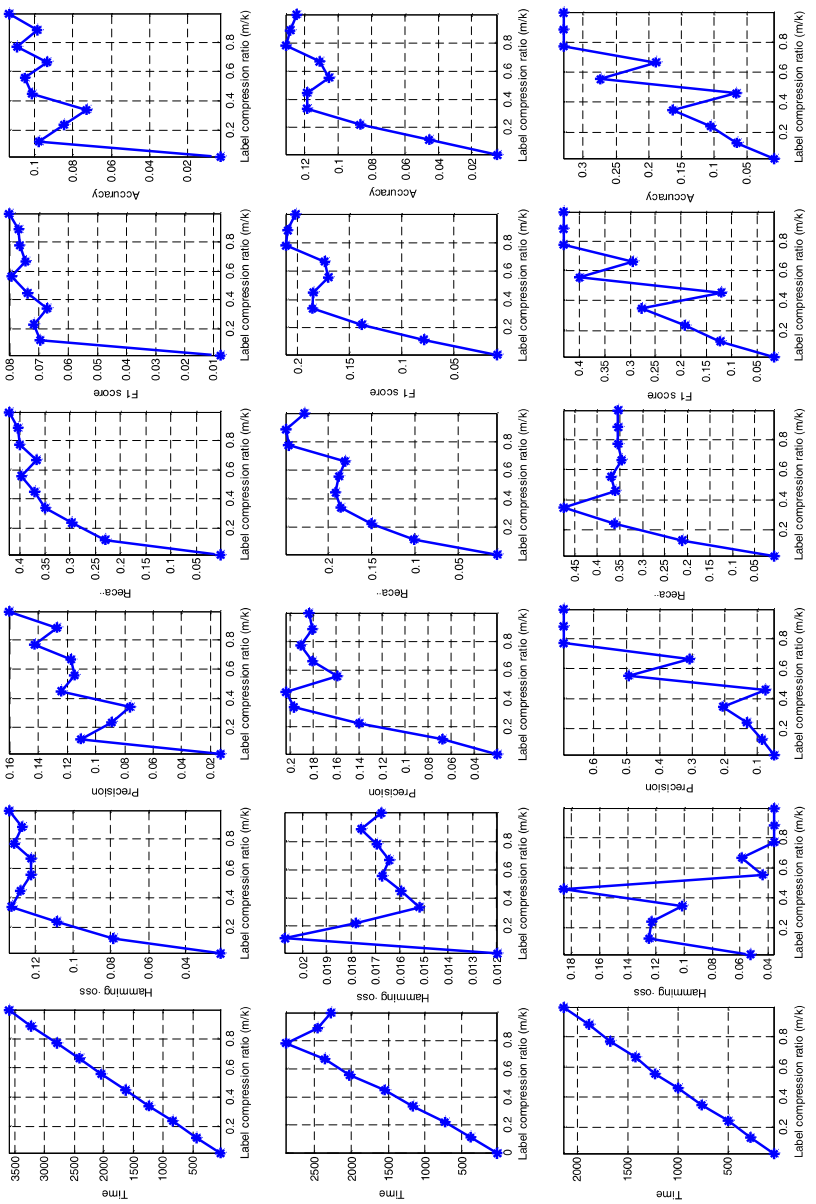

Fig. 7 Trade-off between label compression and 5 prediction performance metrics, time costs on 5 datasets. From *top* to *bottom*: Bibtex, Corel5k, Mediamill, Enron and Medical

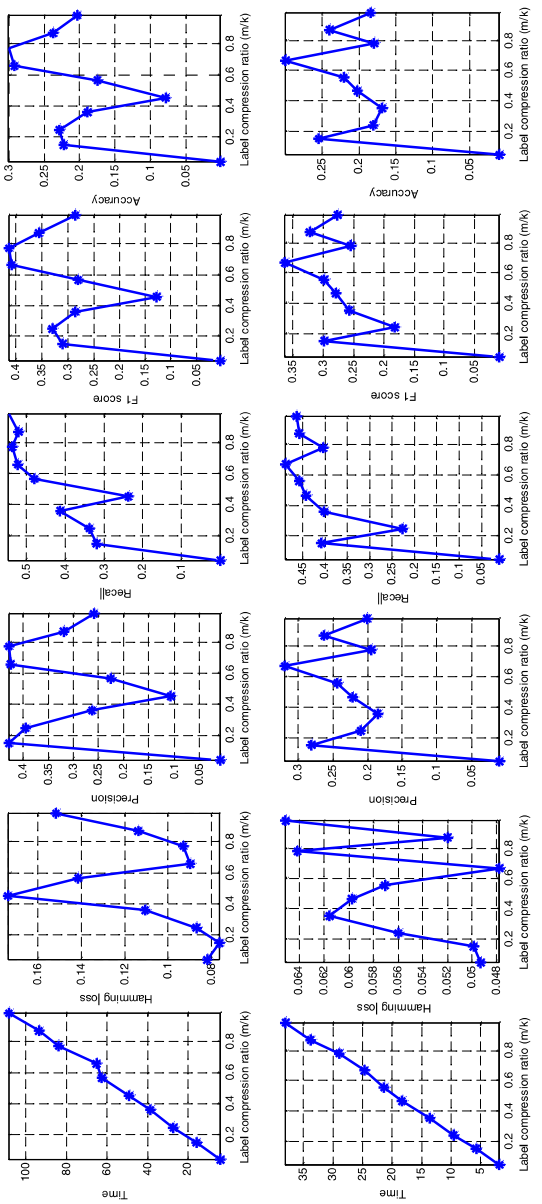

Fig. 7 (*Continued*)

We have developed a greedy binary matrix decomposition method, the "labelset distilling method (LDM)", to extract the discrete patterns, i.e., distilled labelsets (DLs) that are frequently shared by the label vectors in the original label matrix. LDM recursively divides the label vectors in the label matrix into several clusters, extracts the shared label subset of each cluster as a distilled labelset (DL), adds it to the DLs and subtracts it from the label vectors in the corresponding cluster. We have discovered that the random projection signs of an arbitrary DL and a given label vector have an explicit joint distribution in two cases, i.e., the DL is included in the label vector, and the DL is not included in the label vector. The computation of these two kinds of joint distributions is accomplished by a geometric inference.

In the recovery algorithm of CL, the empirical estimation of the joint distribution is calculated from the predicted CL label vector and the random projection signs of distilled labelsets. A series of statistical tests are conducted on the empirical joint distributions of all the DLs to determine whether each DL is included in the original label vector. This test is based on a comparison of the KL divergences between the empirical joint distribution and the two explicit joint distributions in two different cases.

CL solves or at least substantially alleviates the three main problems harassing multi-label learning, which are the problem of sample imbalance, the problem of label dependence and the problem of label high dimensionality. CL is a general method that allows direct embedding of most existing multi-label and single-label learning techniques in its training stage, their advantages will be inherited in their CL variants for multi-label learning. Since CL significantly reduces the problem size by label compression and compressed CL labels can be efficiently recovered, it provides an efficient solver for large-scale multi-label learning tasks. In CL, the label dependence information is stored in DLs after compression and used in the recovery algorithm. LDM is also an isolated method that can be used in other multi-label learning methods to exploit the label dependence. To our best knowledge, CL is the first multi-label learning method with model complexity much less than that of BR, while the label correlations are simultaneously explored.

Theoretically, we have proved that the probabilistic upper bounds of recovery failures exponentially shrink with increasing either the dimensionality of the CL label vector or the cardinality of DL. These analyses demonstrate the effectiveness of label compression and prediction improvement brought by LDM. In the near future, we will theoretically study how the error (or regret) of the learned binary classifiers for the subproblems is transformed into the error (or regret) of the final classifier for the multi-label problem.

We have evaluated the performance of CL in label compression/recovery and multi-label prediction via 5 groups of experiments on 21 datasets from different real-world problems. In the experiments, we have compared CL with BR, 3 multi-label learning methods (ML-knn, MDDM and ML-CS) and 2 SVM algorithms dealing with imbalanced data on 5 prediction performance metrics and time cost. We also have studied the trade-off between compression and performance improvement in CL without using DLs. The experimental results of label compression/recovery have verified our theoretical analyses about the improvement of the sample balance, elimination of label independence and accurate recovery brought by CL. The five groups of multi-label prediction experiments demonstrate the competitive prediction performance, efficiency and robustness of CL in multi-label learning.

Acknowledgements The authors would like to thank the anonymous reviewers who have provided constructive comments on improving this paper. This research is supported in part by the Australia ARC discovery project (ARC DP-120103730) and the US National Science Foundation (NSF CCF-0905337).

References

Bian, W., & Tao, D. (2011). Max-min distance analysis by using sequential sdp relaxation for dimension reduction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *33*(5), 1037–1050.

Bianchi, N.C., Gentile, C., & Zaniboni, L. (2006). Incremental algorithms for hierarchical classification. *Journal of Machine Learning Research*, *7*, 31–54.

Boutell, M.R., Luo, J., Shen, X., & Brown, C.M. (2004). Learning multi-label scene classification. *Pattern Recognition*, *37*(9), 1757–1771.

Breiman, L., & Friedman, J.H. (1997). Predicting multivariate responses in multiple linear regression (with discussion). *The Journal of the Royal Statistical Society Series B*, *54*, 5–54.

Candès, E.J., Romberg, J.K., & Tao, T. (2006). Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, *52*(2), 489–509.

Chang, C.-C., & Lin, C.-J. (2001). LIBSVM: a library for support vector machines. *Software available at* http://www.csie.ntu.edu.tw/~cjlin/libsvm.

Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P. (2002). Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, *16*, 321–357.

Chen, J., Liu, J., & Ye, J. (2010). Learning incoherent sparse and low-rank patterns from multiple tasks. In *SIGKDD'10: The 16th ACM SIGKDD international conference on knowledge discovery and data mining*.

Cheng, W., & Hüllermeier, E. (2009). Combining instance-based learning and logistic regression for multilabel classification. *Machine Learning*, *76*(2–3), 211–225.

Clare, A., & King, R. D. (2001). Knowledge discovery in multi-label phenotype data. In *PKDD'01: Proceedings of the 5th European conference on principles of data mining and knowledge discovery* (pp. 42–53). London: Springer.

Clarkson, K.L. (2008). Tighter bounds for random projections of manifolds. In *SCG'08: Proceedings of the 24 annual symposium on computational geometry* (pp. 39–48).

Crammer, K., & Singer, Y. (2003). A family of additive online algorithms for category ranking. *Journal of Machine Learning Research*, *3*, 1025–1058.

Dasgupta, S. (2000). Experiments with random projection. In *UAI'00: Proceedings of the 16th conference on uncertainty in artificial intelligence*, San Francisco, CA, USA (pp. 143–151).

Dasgupta, S., & Freund, Y. (2008). Random projection trees and low dimensional manifolds. In *STOC'08: Proceedings of the 40th annual ACM symposium on theory of computing* (pp. 537–546).

Dembczyński, K., Cheng, W., & Hüllermeier, E. (2010). Bayes optimal multilabel classification via probabilistic classifier chains. In *The 27th international conference on machine learning* (ICML 2010).

Dembczyński, K., Waegeman, W., Cheng, W., & Hüllermeier, E. (2010). On label dependence in multi-label classification. In *ICML 2010 workshop on learning from multi-label data (MLD 10)* (pp. 5–13).

Dietterich, T.G., & Bakiri, G. (1995). Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research*, *2*, 263–282.

Diplaris, S., Tsoumakas, G., Mitkas, P.A., & Vlahavas, I. (2005). Protein classification with multiple algorithms. In *Proceedings of the 10th Panhellenic conference on informatics (PCI 2005)* (pp. 448–456).

Donoho, D.L. (2006). Compressed sensing. *IEEE Transactions on Information Theory*, *52*(4), 1289–1306.

Duygulu, P., Barnard, K., de Freitas, N., & Forsyth, D. (2002). Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In *ECCV'02: Proceedings of the 7th European conference on computer vision-part IV* (pp. 97–112). London: Springer.

Efron, B., Hastie, T., Johnstone, L., & Tibshirani, R. (2002). Least angle regression. *Annals of Statistics*, *32*, 407–499.

Escalera, S., Pujol, O., & Radeva, P. (2010). On the decoding process in ternary error-correcting output codes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *32*(1), 120–134.

Evgeniou, T., & Pontil, M. (2004). Regularized multi-task learning. In *KDD'04: Proceedings of the tenth ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 109–117).

Fürnkranz, J., Hüllermeier, E., Mencía, E.L., & Brinker, K. (2008). Multilabel classification via calibrated label ranking. *Machine Learning*, *73*(2), 133–153.

Ghamrawi, N., & McCallum, A. (2005). Collective multi-label classification. In *CIKM'05: Proceedings of the 14th ACM international conference on information and knowledge management* (pp. 195–200).

Goemans, M.X., & Williamson, D.P. (1995). Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, *42*(6), 1115–1145.

Gomez, J., Boiy, E., & Moens, M.-F. (2011). Highly discriminative statistical features for email classification. *Knowledge and Information Systems*. doi:10.1007/s10115-011-0403-7.

Gretton, A., Bousquet, O., Smola, E., & Schölkopf, B. (2005). Measuring statistical dependence with Hilbert-Schmidt norms. In *Proceedings algorithmic learning theory* (pp. 63–77). Berlin: Springer.

Guan, N., Tao, D., Luo, Z., & Yuan, B. (2011). Non-negative patch alignment framework. *IEEE Transactions on Neural Networks*, *22*(8), 1218–1230.

Gupta, A., Nowak, R., & Recht, B. (2010). Sample complexity for 1-bit compressed sensing and sparse classification. In *Proceedings of the IEEE international symposium on information theory (ISIT)*.

Halko, N., Martinsson, P.-G., & Tropp, J.A. (2009). Finding structure with randomness: Stochastic algorithms for constructing approximate matrix decompositions. arXiv:0909.4061.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: data mining, inference, and prediction* (2nd ed.). *Springer series in statistics*. Berlin: Springer. Corr. 3rd printing edition.

Hsu, D., Kakade, S.M., Langford, J., & Zhang, T. (2009). Multi-label prediction via compressed sensing. In *Advances in neural information processing systems 23*.

Hüllermeier, E., Fürnkranz, J., Cheng, W., & Brinker, K. (2008). Label ranking by learning pairwise preferences. *Artificial Intelligence*, *172*(16–17), 1897–1916.

Indyk, P., & Motwani, R. (1998). Approximate nearest neighbors: towards removing the curse of dimensionality. In *STOC'98: Proceedings of the thirtieth annual ACM symposium on theory of computing* (pp. 604–613).

Jia, S., & Ye, J. (2009). An accelerated gradient method for trace norm minimization. In *The 26th international conference on machine learning (ICML)* (pp. 457–464).

Jia, S., Tang, L., Yu, S., & Ye, J. (2010). A shared-subspace learning framework for multi-label classification. *ACM Transactions on Knowledge Discovery from Data*, 2(1).

Johnson, W., & Lindenstrauss, J. (1984). Extensions of Lipschitz mappings into a Hilbert space. In *Contemporary Mathematics: Vol. 26. Conference in modern analysis and probability*, New Haven, Conn., 1982 (pp. 189–206). Providence: Am. Math. Soc.

Katakis, I., Tsoumakas, G., & Vlahavas, I. (2008). Multilabel text classification for automated tag suggestion. In *Proceedings of the ECML/PKDD 2008 discovery challenge*.

Kong, X., & Yu, P. (2011). gmlc: a multi-label feature selection framework for graph classification. *Knowledge and Information Systems*. doi:10.1007/s10115-011-0407-3.

Koufakou, A., Secretan, J., & Georgiopoulos, M. (2011). Non-derivable itemsets for fast outlier detection in large high-dimensional categorical data. *Knowledge and Information Systems*, *29*, 697–725.

Kullback, S., & Leibler, R.A. (1951). On information and sufficiency. *The Annals of Mathematical Statistics*, *22*(1), 79–86.

Langford, J., & Beygelzimer, A. (2005). Sensitive error correcting output codes. In *COLT'05: annual conference on learning theory* (Vol. *3559*, pp. 158–172).

Li, P. (2008). Estimators and tail bounds for dimension reduction in $\ell_\alpha (0 < \alpha \leq 2)$ using stable random projections. In *SODA'08: Proceedings of the nineteenth annual ACM-SIAM symposium on discrete algorithms* (pp. 10–19). Philadelphia: Society for Industrial and Applied Mathematics.

Li, P. (2010). Approximating higher-order distances using random projections. In *The 26th conference on uncertainty in artificial intelligence* (UAI 2010).

Liu, L., & Liang, Q. (2011). A high-performing comprehensive learning algorithm for text classification without pre-labeled training set. *Knowledge and Information Systems*, *29*, 727–738.

Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and Computing*, *17*(4), 395–416.

MacQueen, J.B. (1967). Some methods for classification and analysis of multivariate observations. In *Proceeding of the fifth Berkeley symposium on mathematical statistics and probability* (Vol. 1, pp. 281–297).

Masud, M., Woolam, C., Gao, J., Khan, L., Han, J., Hamlen, K., & Oza, N. (2011). Facing the reality of data stream classification: coping with scarcity of labeled data. *Knowledge and Information Systems*. doi:10.1007/s10115-011-0447-8.

Mencía, L., & Fürnkranz, J. (2008). Pairwise learning of multilabel classifications with perceptrons. In *IEEE international joint conference on neural networks (IJCNN-08)* (pp. 995–1000).

Ng, A.Y., Jordan, M.I., & Weiss, Y. (2001). On spectral clustering: Analysis and an algorithm. In *NIPS'01: advances in neural information processing systems 14* (Vol. 2, pp. 849–856).

Osuna, E., Freund, R., & Girosi, F. (1997). *Support vector machines: Training and applications* (Technical report). Massachusetts Institute of Technology.

Raginsky, M., & Lazebnik, S. (2009). Locality-sensitive binary codes from shift-invariant kernels. In *The 23rd annual conference on neural information processing systems (NIPS 2009)*.

Read, J. (2010). Meka softwares.

Read, J., Pfahringer, B., & Holmes, G. (2008). Multi-label classification using ensembles of pruned sets. In *ICDM'08: Proceedings of the 2008 eighth IEEE international conference on data mining*, Washington, DC, USA (pp. 995–1000).

Read, J., Pfahringer, B., Holmes, G., & Frank, E. (2009). Classifier chains for multi-label classification. *Machine Learning*, *85*(3), 333–359.

Schapire, R.E., & Singer, Y. (2000). Boostexter: a boosting-based system for text categorization. *Machine Learning*, *39*(2/3), 135–168.

Si, S., Tao, D., & Geng, B. (2010). Bregman divergence-based regularization for transfer subspace learning. *IEEE Transactions on Knowledge and Data Engineering*, *22*(7), 929–942.

Snoek, C. G. M., Worring, M., van Gemert, J. C., Geusebroek, J. M., & Smeulders, A. W. M. (2006). The challenge problem for automated detection of 101 semantic concepts in multimedia. In *MULTIMEDIA '06: Proceedings of the 14th annual ACM international conference on multimedia* (pp. 421–430). New York: ACM.

Tao, D., Li, X., Wu, X., & Maybank, S.J. (2009). Geometric mean for subspace selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *31*(2), 260–274.

Trohidis, K., Tsoumakas, G., Kalliris, G., & Vlahavas, I. (2008). Multilabel classification of music into emotions. In *Proc. 9th international conference on music information retrieval (ISMIR 2008)*. Philadelphia, PA, USA.

Tsoumakas, G. (2010). Mulan: A java library for multi-label learning. http://mulan.sourceforge.net/.

Tsoumakas, G., & Katakis, I. (2007). Multi-label classification: An overview. *International Journal of Data Warehousing and Mining*, *3*(3), 1–13.

Tsoumakas, G., Katakis, I., & Vlahavas, I. (2008). Effective and efficient multilabel classification in domains with large number of labels. In *Proc. ECML/PKDD 2008 workshop on mining multidimensional data (MMD'08 2007)*.

Tsoumakas, G., Katakis, I., & Vlahavas, I. (2010). Mining multi-label data. In *Data mining and knowledge discovery handbook*.

Tsoumakas, G., & Vlahavas, I. (2007). Random k-labelsets: An ensemble method for multilabel classification. In *Proceedings of the 18th European conference on machine learning (ECML 2007)*, Warsaw, Poland, 2007 (pp. 406–417)

Ueda, N., & Saito, K. (2002). Parametric mixture models for multi-labeled text. In *Advances in neural information processing systems 15. Neural information processing systems, NIPS 2002*.

Vapnik, V. N. (1995). *The nature of statistical learning theory*. New York: Springer.

Vempala, S.S. (2004). *The random projection method. DIMACS Series in Discrete Mathematics and Theoretical Computer Science: Vol. 65*. Providence: Am. Math. Soc.

Yates, F. (1934). Contingency tables involving small numbers and the χ^2 test. *Journal of the Royal Statistical Society*, *1*, 217–235.

Zhang, M., & Wang, Z. (2009). Mimlrbf: Rbf neural networks for multi-instance multi-label learning. *Neurocomputing*, *72*(16–18), 3951–3956.

Zhang, M., & Zhou, Z. (2006). Multi-label neural networks with applications to functional genomics and text categorization. *IEEE Transactions on Knowledge and Data Engineering*, *18*(10), 1338–1351.

Zhang, M., & Zhou, Z. (2007). Ml-knn: A lazy learning approach to multi-label learning. *Pattern Recognition*, *40*(7), 2038–2048.

Zhang, Y., & Zhou, Z. (2008). Multi-label dimensionality reduction via dependence maximization. In *AAAI'08: Proceedings of the 23rd national conference on artificial intelligence* (pp. 1503–1505).

Zhou, T., Tao, D., & Wu, X. (2010). Nesvm: A fast gradient method for support vector machines. In *ICDM'10: Proceedings of the 2010 IEEE international conference on data mining* (pp. 679–688).

Zhou, T., Tao, D., & Wu, X. (2011). Manifold elastic net: a unified framework for sparse dimension reduction. *Data Mining and Knowledge Discovery (Springer)*, *22*(3), 340–371.