Multi-label Learning with Highly Incomplete Data via Collaborative Embedding

Yufei Han
Symantec Research Labs
yufei_han@symantec.com

Guolei Sun
KAUST
guolei.sun@kaust.edu.sa

Yun Shen
Symantec Research Labs
yun_shen@symantec.com

Xiangliang Zhang
KAUST
xiangliang.zhang@kaust.edu.sa

ABSTRACT

Tremendous efforts have been dedicated to improving the effectiveness of multi-label learning with incomplete label assignments. Most of the current techniques assume that the input features of data instances are complete. Nevertheless, the co-occurrence of highly incomplete features and weak label assignments is a challenging and widely perceived issue in real-world multi-label learning applications due to a number of practical reasons including incomplete data collection, moderate labels from annotators, etc. Existing multi-label learning algorithms are not directly applicable when the observed features are highly incomplete. In this work, we attack this problem by proposing a weakly supervised multi-label learning approach, based on the idea of *collaborative embedding*. This approach provides a flexible framework to conduct efficient multi-label classification at both transductive and inductive mode by coupling the process of reconstructing missing features and weak label assignments in a joint optimisation framework. It is designed to collaboratively recover feature and label information, and extract the predictive association between the feature profile and the multi-label tag of the same data instance. Substantial experiments on public benchmark datasets and real security event data validate that our proposed method can provide distinctively more accurate transductive and inductive classification than other state-of-the-art algorithms.

CCS CONCEPTS

• **Theory of computation → Inductive inference**; **Semi-supervised learning**;

KEYWORDS

Multi-label learning;Highly incomplete feature;Weak labels

ACM Reference Format:
Yufei Han, Guolei Sun, Yun Shen, and Xiangliang Zhang. 2018. Multi-label Learning with Highly Incomplete Data via Collaborative Embedding . In *KDD'18: The 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, August 19–23, 2018, London, United Kingdom.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3219819.3220038

KDD'18, August 19–23, 2018, London, United Kingdom

ACM ISBN 978-1-4503-5552-0/18/08…$15.00
https://doi.org/10.1145/3219819.3220038

1 INTRODUCTION

In real-world practices of multi-label learning where data are collected from cybersecurity systems and distributed sensor network, highly incomplete training information is a challenging, while frequently witnessed issue, due to on-device privacy control, limited coverage of deployed sensors and unexpected sensor failures. Furthermore, annotating the collected data become extremely difficult when high fractions of data missing. Human annotators can only provide class labels with high confidence to data samples with well defined feature representation, while leaving the ambiguous data samples unlabelled. Multi-label learning in such a scenario faces difficulties of the co-occurrence of **weak positive unlabelled class tag assignment** and **missing feature values**.

We target on solving this multi-label learning problem with **highly incomplete data**. Deriving stable and accurate estimation of multi-label class membership under the weakly supervised scenario remains a more difficult task yet to be resolved, compared to the problem in a traditional semi-supervised learning setting, where both positive and negative labels are visible. A concrete example of co-occurrence of incomplete feature profiles and weak label assignments is the malicious event categorisation in IoT security, e.g., detecting malicious events on distributed IoT devices. Security telemetry data collected from security products protecting IoT devices, e.g., IDS-enabled routers or network firewalls/proxy servers, can be highly incomplete. Customers can also manually change their privacy configurations on their devices to limit coverage of telemetry data shared with security vendors. Simultaneously, human analysts can only identify malicious events featuring with relatively complete and typical profiles since mislabeling malicious events as benign ones can potentially increase false negative rate and make learnt event detectors fail to capture potential malicious activities. In this cost-sensitive scenario, it is sensible for human analysts to leave the security events with incomplete and ambiguous profiles unlabeled. An efficient solution to multi-label learning with the co-occurrence of **missing feature values** and **weak label assignments** is therefore essential to the success of data analytics tasks in real-world applications like IoT security.

Existing methods either address the problem of weak label assignment [2, 32] or focus on the noisy feature values [3, 6, 11], while not solving the learning problem with co-occurrence of partially

observed feature and weak label information. In [2, 32], multi-label learning with weak label information is studied with the assumption that the feature description of data instances is fully observed and free from corruption. Multi-label learning with missing information in feature and label is studied in [11] and [3]. However, the solutions are not designed to handle the positive-unlabeled characteristic of the weak label assignments. They require both positive and negative label elements observed in the given label annotations for training. In [6], Chiang *et al.* introduces a dirty data model to improve the robustness of multi-label learning when having complete but noise-corrupted features. Note that this noise model is designed to characterize noise distribution. When facing a large fraction of missing features, we can barely have any prior assumption on the missing values and their statistical properties. This makes it difficult to apply the noise-tolerant learning methods [6] directly as a solution in our work.

In this paper, we propose Collaborative Embedding (**ColEmbed**) to attack the issue of the co-occurrence of highly incomplete feature profiles and weak labels in multi-label learning. Firstly we learn collaboratively low-rank approximation to incomplete features and weak labels to derive the shared embedding based representation of features and labels. Given the positive-unlabeled intrinsic of the weak label matrix, we adopt a cost-sensitive matrix factorization to tune the label embeddings to better recover class distribution. By constraining the consistency between feature and label embedding space, the discriminative information extracted from incomplete feature profiles helps to recover label assignments. Notably, benefited from the flexible algorithmic design, the proposed **ColEmbed** method can perform both transductive and inductive multi-label learning, with wide applicability. The evaluation results on both public benchmark datasets and a real-world IoT security data show that the proposed method has superior performances in both transductive and inductive learning scenarios, compared to the state-of-the-art multi-label learning solutions.

2 RELATED WORK

The problem of multi-label learning with weak labels was firstly defined in [32], where the learning process was considered as a semi-supervised learning problem. The proposed solution, named **WELL**, investigated the low-rank structure of the pairwise affinity matrix of training data instances. Intuitively, if a given pair of data instances are assigned a high affinity score, it is likely that the two data instances share the same label assignment. Based on this assumption, correlation between label dimensions is then recovered to estimate missing label assignments. When a high fraction of feature profiles are missing, the pairwise affinity relation and label correlation cannot be estimated stably, and thus cause distinctive deterioration of multi-label learning performances.

The problem of learning with weak labels was casted as Positive-Unlabeled (PU) learning in [4, 9, 10, 15, 23]. In PU learning, only a small fraction of data instances from positive class are well labeled, while labels of the rest data are not available. This weakly supervised learning scenario is similar with the setting of the weak label problem. By treating each label dimension independently, multi-label learning with weak labels can be decomposed into a series of parallel PU learning tasks. However, ignoring the correlation information in the primitive solution does harm to classification performances, as stated in [27, 32]. To model the correlation among labels, the whole PU labeled matrix was taken as input in [13], and processed by a low-rank cost-sensitive matrix factorization to achieve completion of the given PU matrix. The proposed PU matrix completion has an inductive extension to employ features of instance as inductive side information. Nevertheless, it assumes the feature observations are complete and free from corruption, which limits its capability on solving our studied problem.

Another highly relevant topic is semi-supervised multi-label classification with incomplete labels. In this problem, both positive and negative label entries are provided. Early stage research of this topic conducted label imputation as a preprocessing step [16, 17, 28, 32, 34]. They don't explicitly employ predictive relation between feature and labels, which limits their effectiveness. Recent works overcame this shortage by jointly exploiting low rank structures of label assignments and predicative relation between features and labels, as reported in [3, 6, 11, 12, 29–31, 33]. Enforcing the low rank constraint on label assignments is testified to be efficient to capture latent correlation between labels. Furthermore, features were employed as predicative side information to reconstruct missing labels recovery [11, 29, 30, 33]. The semi-supervised setting includes the weak label based learning as a special case. However, the positive-unlabeled intrinsic of the weak label issue introduces more bias into the estimates of classification distribution, thus increases difficulty of learning.

It is worth noting that [3, 6, 11] accepted imperfect features as input. [11] and [3] assumed the entries of both feature and label matrix were randomly selected as missing observations. They jointly reconstructed features and labels by performing low-rank matrix completion on the augmented matrix concatenated by both features and labels. Nevertheless, both positive and negative class labels are required in both algorithms to identify the potential class separating structure. With only positive labels observed in the weak label problem, the two approaches can be severely biased from the true class assignments. Furthermore, they don't enforce the predictive correlation and information consistency between features and labels. [6] assumed the features were fully observed but noise-corrupted. When the fraction of missing features is high (e.g., over 50%), the low signal-noise ratio can weaken robustness of the method and deteriorate its label reconstruction performance.

Motivated by the previous efforts, we propose to conduct collaborative learning of low-rank approximation to incomplete features and weak labels. The key idea is to approximate the underlying predicative relation between feature space and label embedding space, in order to conduct both transductive and inductive multi-label classification with highly incomplete training information. This approach is inspired by the natural learning loop of human beings - transductive and inductive learning process are usually coupled and conducted in parallel. The former unveils the associations between feature profiles and multi-label tags, while the latter deduces a reusable classification model to categorize new data instances. The success of the state-of-the-art approaches [6, 13, 31] confirms the merit of introducing the flexibility of combing transductive and inductive learning together. Nevertheless, they become fragile with the co-occurrence of missing features and weak labels.

3 THE PROPOSED COLLABORATIVE EMBEDDING METHOD

In this section, we first define the studied problem and then introduce a provably way for label matrix reconstruction in Section 3.2. After that, we propose a general model for collaborative learning with incomplete data in Section 3.3 and its linear and non-linear implementation in Section 3.4.

3.1 Notations and Problem Definition

Given N instances as training data, we use $X \in \mathbb{R}^{N*D}$ to denote D-dimensional feature vectors of all N instances. X is partially observed, with a binary-valued matrix Ω indicating locations of miss-observations in X. $\Omega_{i,j} = 1$ if $X_{i,j}$ is observed, otherwise $\Omega_{i,j} = 0$. $Y \in \{0,1\}^{N*M}$ denotes the label assignments of the training instances. M is the label dimension. In the setting of the weak label problem, Y is intrinsically a positive-unlabeled matrix, $Y_{i,j} = 1$ if $Y_{i,j}$ is observed. More precisely, $Y_{i,j} = 1$ indicates that the corresponding instance $X_{i,:}$ is positively labeled in the j-th label dimension. $Y_{i,j} = 0$ if $Y_{i,j}$ is not observed, thus categorized as unlabeled elements in Y.

In this work, the proposed algorithm is required to first conduct **transductive multi-label learning**, which aims at accurately recovering the missing label elements of Y based on highly incomplete X. In a further step, the algorithm is also designed to perform **inductive learning**. Based on incomplete feature profile X and weak labels Y, the proposed algorithm builds a reusable classifier f that can be applied on new testing instances of fully observed profiles for classification use. In the following sections, without extra highlight, $\|\cdot\|$ denotes Frobenius norm of a matrix. $\circ$ is used to denote the Hadamard product operator, such as $[X \circ Y]_{i,j} = X_{i,j}Y_{i,j}$. $\|\cdot\|_*$ denotes the nuclear norm of X.

3.2 Provably Reconstruction of Label Matrix

We cast the problem of learning with incomplete features X and weak labels Y as a positive-unlabeled (PU) learning process. The basic idea originates from cost-sensitive matrix factorization proposed in [13]. Assuming that the underlying complete true features of instances are given as X^{gt}, we propose to recover the underlying true label assignments based on logistic matrix factorization, which encodes label correlation. The objective function of label reconstruction is given in Eq. (1):

$$S^* = \underset{S}{arg\,min} \sum_{i,j} \Gamma_{i,j} \log(1 + e^{(1-2Y_{i,j})X^{gt}_{i,:}(S_{j,:})^T}) + \lambda_S \|S\|_* \quad (1)$$

where $\Gamma_{i,j}$ is an element-wise calibrating parameter reweighing the mis-classification cost over positive entries and unlabeled entries in Y. It outputs $\alpha \in [0,1]$ when $Y_{i,j} = 1$. For unlabeled $Y_{i,j}$, it is valued as $1 - \alpha$. The reweighing scheme originates from the idea of cost-sensitive calibration for learning with noisy labels [19, 21], which is widely accepted in PU learning. λ_S enforces the low rank constraint on S, which in turn strengthens the low-rank structure of the reconstruction to Y. Since the correlation exits between label dimensions in most real-world multi-label tags, requiring the recovered Y to be low rank is helpful for better estimating the missing label assignments.

The probabilistic class membership of each element $Y_{i,j}$ is estimated by $\hat{Y} = 1/(1 + exp(-X^{gt}_{i,:}(S_{j,:})^T))$. Binary labels are inferred by thresholding $\hat{Y}$ as $\overline{Y}_{i,j} = I(\hat{Y}_{i,j} > q)$. $I()$ is the indicator function and $q \in \mathbb{R}$ is the threshold. Without loss of generality, we set $q = 0.5$ in the following theoretical study. Otherwise we can shift and scale the matrix instead. According to co-embedding theory [12, 18], $S \in \mathbb{R}^{M*D}$ defines a linear transformation mapping a data instance from the original feature space to the label space. The dot product between $X^{gt}_{i,:}$ and $S_{j,:}$ measures the association strength between each instance and label [12, 18].

Let Y^{gt} be the underlying true labels. We can provide an analytical upper bound of the empirical reconstruction error between $\overline{Y}$ and Y^{gt} as follows. Assuming X^{gt} provides perfect side information, a.k.a. $Col(Y^{gt}) \subseteq Col X^{gt}$, the upper bound of the reconstruction error can be derived following the proof of Theorem 3 in [13]:

Theorem 1. Assume $Col(Y^{gt}) \subseteq Col(X^{gt})$, where $Col()$ denotes the column space. α is valued as $\frac{1+\rho}{2}$ and ρ is the flipping rate of masking positively labeled entries of Y as unlabeled. $\hat{Y}^*$ is derived with S^*, the minimiser of Eq. (1). Let $\overline{Y}^*_{i,j}$ be the thresholded 0-1 matrix of $\hat{Y}^*$. The label reconstruction error is defined as $R(\overline{Y}^*) = \frac{1}{NM}\sum_{i,j} \|\overline{Y}^*_{i,j} - Y^{gt}\|^2$. Then with a constant C, we can derive the following upper bound of the reconstruction error R holding with probability at least $1 - \delta$:

$$R(\overline{Y}^*) \leq \frac{12}{1-\rho}\left(\frac{Ct\sqrt{\log 2D}}{\sqrt{NM}}\varkappa + 2\frac{\sqrt{\log(2\delta)}}{\sqrt{2NM}}\right) \quad (2)$$

where t is the upper bound of the spectral norm of H, and $\varkappa = \max_i \|X^{gt}_{i,:}\|$ is the maximum $L2$-norm of the row vector in X^{gt}. Note that $\varkappa$ is a bounded constant since X^{gt} in practice is always bounded. As we can find, with good side information X^{gt}, the average reconstruction error of binary label reconstruction is of the order of $O(1/(N(1-\rho)))$. However in our case, we are not given with X^{gt}, but the incomplete feature matrix X, where only a small fraction of the truth feature values are observed. The incompleteness of X violates the assumption on the quality of side information in **Theorem 1**. Therefore we next propose a new model for learning with the incomplete X.

3.3 Collaborative Learning with Incomplete Training Data

Despite incomplete, the association existing between instances and feature dimensions can be helpful for estimating affinity relation of data instances, which is closely relevant with the distribution of label assignments in the label embedding space. Following this principle, we propose to jointly recover the missing observations of feature profiles and labels, in order to conquer the challenging issue introduced by the incomplete training data. In Eq. (3), we define the learning objective of the proposed Collaborative Embedding method (**ColEmbed**):

$$\begin{aligned} \hat{X}^*, f^*, E^* = \underset{\hat{X},f,E}{arg\,min} &\|\Omega \circ (X - \hat{X})\|^2 + \lambda\ L(\hat{X}, Y, f, E) \\ &+ \lambda_E\|E\|_* + \lambda_X\|\hat{X}\|_* + \lambda_f L_{reg}(f) \end{aligned} \quad (3)$$

$$\text{with}\ \ L(\hat{X}, Y, f, E) = \sum_{i,j} \Gamma_{i,j} \log(1 + e^{(1-2Y_{i,j})(f(\hat{X}_{i,j}) + E_{i,j})})$$

The merits of the above objective function that jointly learns a low-rank approximation $\hat{X}$ to X and a linear predictor $f(\hat{X}) + E$ to recover weak labels are **three-folds**. **Firstly**, the low-rank structure of $\hat{X}$ encodes affinity relation between data instances. Label assignments can make use of the affinity relation to better estimate missing label tags belonging to associated instances or label dimensions. **Secondly**, the recovered label assignments can be considered as an additional regularization for recovering missing feature profiles, as similar feature profiles usually indicate similar label tagging and vice versa. Consequently, the learning processes of recovering features and labels are designed to oscillate iteratively between two ends to share complementary information between each other, until the learning processes converges to a balancing point. Parameter λ weights the relative importances of reconstructing the feature matrix X and recovering the incomplete label matrix Y. Parameter λ_X is the penalty parameter enforcing the low-rank constraint on the approximation of $\hat{X}$. **Finally**, we use $f(\hat{X})$ as a generalised feature extraction procedure, generating new feature representation from the approximation $\hat{X}$. Learning the structure of f is coupled with learning of missing features and labels, which makes f better fit to the underlying predictive relation between features and labels given highly incomplete data. $L_{reg}(f)$ denotes the regularisation on f to prevent the risk of overfitting. E is introduced as the residual error outside the feature space defined by $f(\hat{X})$. It is used to capture the information in the label reconstruction process that $f(\hat{X})$ fails to describe. E is also assumed to be low-rank (penalized with λ_E), in order to preserve the overall low-rank structure of the recovered label matrix. According to the dirty data model proposed in [6], introducing E as a bias term in the inductive learning model improves its robustness against the artifacts in the low-rank approximation $\hat{X}$. In our work, E is used to suppress the impact of the label reconstruction error propagated during iteratively updating the estimation of missing features and labels.

For **transductive learning**, the optimally tuned f, as well as E, represent the side information to recover the missing label and feature elements in the given incomplete data. For **inductive learning**, f is used as a learnt classification model. Performing f on newly observed testing instances with completed feature profiles predicts directly labels of the testing instances. In the followings, we study the impact of different formulations of f by investigating how **linear and non-linear** f perform in the learning task.

3.4 Linear and Non-linear Collaborative Embedding

Firstly, we define $f(\hat{X}) = \hat{X}S^T$ as a linear projection of $\hat{X}$, where $S \in \mathbb{R}^{M*D}$. The regularization term $L_{reg}(f)$ is reduced to $\|S\|_*$, in order to preserve the low-rank structure of the predicted label assignments. As a result, the loss function of the **linearized ColEmbed** method (named **ColEmbed-L**) is given as:

$$\begin{aligned}\hat{X}^*, S^*, E^* = \underset{\hat{X},S,E}{arg\,min} &\|\Omega \circ (X - \hat{X})\|^2 + \lambda L(\hat{X}, Y, S, E) \\ &+ \lambda_E\|E\|_* + \lambda_X\|\hat{X}\|_* + \lambda_S\|S\|_*\end{aligned} \tag{4}$$

$$\text{with } L(\hat{X}, Y, S, E) = \sum_{i,j} \Gamma_{i,j} \log(1 + e^{(1-2Y_{i,j})(\hat{X}_{i,:}(S_{j,:})^T + E_{i,j})})$$

By fixing $\hat{X}$ and discarding the terms irrelevant with S or E, minimizing Eq. (4) with respect to S and E equals to minimize $L(\hat{X}, Y, S, E)$ with $\hat{X}$ as the noisy side information. The process of learning S and E can be considered as inductive multi-label learning with weak labels. Following the proof of Lemma 2 and Theorem 1 in [6], it is straightforward to show that the expected loss $L(\hat{X}, Y, S, E)$ is upper-bounded by the quality of $\hat{X}$ following the proof in [6]:

Theorem 2 Let $\mathcal{N}$ be the trace norm of E, C and $C^{'}$ be universal constants, δ be a constant in $[0, 1]$. $|Y|$ be the cardinality of the set of observed label elements in Y. With probability at least $1 - \delta$, the upper bound of the expected loss $L(\hat{X}, Y, S, E)$ holds:

$$\begin{aligned}L(S, E) \leq min\{&4\mathcal{N}\sqrt{\frac{\log 2N}{|Y|}}, \sqrt{36C\mathcal{B}\frac{\mathcal{N}\sqrt{N}}{|Y|}}\} \\ &+ \frac{4\hat{d}}{C^{'}\mu^2\gamma^2}\sqrt{\frac{\log 2D}{|Y|}} + \mathcal{B}\sqrt{\frac{log\frac{1}{\delta}}{2|Y|}}\end{aligned} \tag{5}$$

In Eq. (5), we inherit the definition of *μ-informative* part of $\hat{X}$ in [6] to measure the quality of the side information $\hat{X}$. It is defined as $\hat{X}_\mu = \sum_{i=1} \sigma_1 I_\mu(\sigma_i/\sigma_1) u_i v_i^T$, where $I_\mu(x)$ is the thresholding operator whose output is x when $x > \mu$ and 0 otherwise. $\hat{X} = \sum_{i=1}^{D} \sigma_i u_i v_i^T$ is the reduced SVD of $\hat{X}$. $\hat{d}$ denotes the rank of the *μ-informative* part of $\hat{X}$. γ is defined as $min_i\|X_{i,:}\|/max_i\|X_{i,:}\|$. $\mathcal{B}$ is the maximum magnitude of the weighted cross-entropy function L with respect to S.

The analytical bound in Eq. (5) illustrates that the side information extracted from the original feature space plays an important role on reducing label inference error. Following this idea, we propose to define a **non-linear** feature extraction process f to improve the descriptive power of side information. An intuitive choice is to apply kernel tricks to build f. However, there are two major barriers of employing kernel machines. Firstly non analytical form of exact kernel mapping can be given to describe new feature representation in the high dimensional kernel space. Thus it is difficult to conduct optimization within the kernel space. Secondly, learning with exact kernel machines is computationally intense.

To address these issues, we use random feature expansion (RFE) based approximation to generate non-linear representation of data instances. RFE method [22] has been used widely to approximate shift-invariant kernel by spanning a randomized feature space based on Fourier transform of the corresponding kernel function. Inner products of two data instances X_i and X_j in the new random space approximates the kernel function $K(X_i, X_j)$. Computationally efficient as it is, RFE can be also considered as non-linear features extracted from the original feature space, and has been used for better inductive matrix completion, as reported in [24]. For constructing **non-linearized ColEmbed** (named **ColEmbed-NL**), we define f as a linear combination of the random features extracted from $\hat{X}$,

$$\begin{aligned}f(\hat{X}) &= \varphi(\hat{X})S^T \quad \text{where} \\ \varphi(\hat{X}_{i,:}) &= \frac{1}{K}[cos(u_1^T\hat{X}_{i,:}), cos(u_2^T\hat{X}_{i,:}), \cdots, cos(u_K^T\hat{X}_{i,:}), \\ &\quad sin(u_1^T\hat{X}_{i,:}), sin(u_2^T\hat{X}_{i,:}), \cdots, sin(u_K^T\hat{X}_{i,:})]\end{aligned} \tag{6}$$

In Eq. (6), $u_1, u_2, ...u_K$ are the K projection directions sampled according to the distribution defined from the Fourier transform of

RBF kernel $k(x_i, x_j) = e^{-\gamma \|x_i - x_j\|^2}$. The distribution is a Gaussian distribution given as $p(u) = N(0, 2\gamma I)$, with γ as the kernel width of RBF kernel. With $S \in \mathbb{R}^{M*K}$ in Eq. (6) as a linear projection matrix, f in **ColEmbed-NL** is defined as a linear combination of the projected non-linear random features of $\hat{X}_{i,:}$. Despite the simplicity, linear combination of the random features generated from RFE can approximate supervised Kernel machines on large-scale data, and provide competitive accuracy with the standard Kernel SVM and achieve much faster training [22]. It also produces very compact functions because only S and a set of $O(K)$ random features need to be retained in the optimization.

Combining Eq. (3) and Eq. (6), the objective function of **ColEmbed-NL** is thus given in Eq. (7):

$$\begin{aligned} \hat{X}^*, S^*, E^* = \underset{\hat{X},S,E}{arg\,min} & \|\Omega^X \circ (X - \hat{X})\|^2 + \lambda L(\hat{X}, Y, S, E) \\ & + \lambda_E \|E\|_* + \lambda_X \|\hat{X}\|_* + \lambda_S \|S\|_* \end{aligned} \tag{7}$$

$$\text{with } L(\hat{X}, Y, S, E) = \sum_{i,j} \Gamma_{i,j} \log(1 + e^{(1-2Y_{i,j})(\varphi(\hat{X}_{i,:}(S_{j,:})^T + E_{i,j})})$$

4 SGD BASED ALTERNATING UPDATE FOR PARAMETER ESTIMATION

Although the loss function defined in Eq. (3) (for both **ColEmbed-L** and **ColEmbed-NL**) is not jointly convex, the convexity holds by optimising with respect to $\hat{X}$ by fixing S and E, or the other way round. Therefore, we propose an efficient alternating update based algorithm to solve Eq. (3) by decomposing it into two sub-problems, as summarised in Algorithm 1. To scale the algorithm on large datasets, we use stochastic gradient descent (SGD) to conduct the alternating optimisation with respect to each variable.

It is known that $\underset{X}{minf}(X) + \lambda |X|_*$ is equivalent to $\underset{U,V}{minf}(UV^T) + \lambda(\|U\|^2 + \|V\|^2)$ when k for $U \in \mathbb{R}^{N*k}$ and $V \in \mathbb{R}^{M*k}$ is sufficiently large. We thus replace $\hat{X}$ as UV^T, S as PQ^T and E as MN^T in the optimization process, with k_X, k_S and k_E columns in U and V, P and Q and M and N, respectively. Consequently, the optimization problem becomes:

$$\begin{aligned} U^*, V^* &= \underset{U,V}{arg\,min} \|\Omega^X \circ (X - UV^T)\|^2 \\ &+ \lambda L(U, V, Y, P^{t-1}(Q^{t-1})^T, G^{t-1}(H^{t-1})^T) + \frac{\lambda_X}{2}(\|U\|^2 + \|V\|^2) \\ P^*, Q^*, G^*, H^* &= \underset{P,Q,G,H}{arg\,min}\ L(U^{t-1}, V^{t-1}, Y, P, Q, G, H) \\ &+ \frac{\lambda_S}{2}(\|P\|^2 + \|Q\|^2) + \frac{\lambda_E}{2}(\|G\|^2 + \|H\|^2) \end{aligned} \tag{8}$$

Based on Eq. (8), we demonstrate the optimisation of the loss function of **ColEmbed-L** with respect to U, P and G in Eq. (9). We slightly abuse the definition of sign function, $sgn(a) = 1$ if a is positive and -1 otherwise. Specially, we conduct SGD with respect to each row vector of U, P and G independently of the other rows.

Algorithm 1: Alternating Update Using SGD

Input: Incomplete feature matrix **X**, weak label matrix **Y**
Output: U,V,P,Q,G,H
Initialize U,V,G,H,P,Q as random valued matrices
for $t = 1$ **to** T **do**
  U^t, V^t = ADAM($P^{t-1}, Q^{t-1}, G^{t-1}, H^{t-1}$)
  P^t, Q^t, G^t, H^t = ADAM(U^t, V^t)
end

As a result, each row vector is updated as follows:

$$\begin{aligned} U_{i,:}^{t+1} &= U_{i,:}^t - \eta\Big(\sum_{j \in \Omega_{i,:}^X} T_{i,j} V_{j,:} + \lambda_X U_{i,:}^t \\ &- \lambda \sum_j \Gamma_{i,j} sgn(Y_{i,j}) \frac{e^{-sgn(Y_{i,j})(U_{i,:}^t V^T (S_{j,:})^T + E_{i,j})}}{1 + e^{-sgn(Y_{i,j})(U_{i,:}^t V^T (S_{j,:})^T) + E_{i,j})}} S_{j,:} V\Big) \\ P_{i,:}^{t+1} &= P_{i,:}^t - \eta(\lambda_H P_{i,:}^t \\ &- \lambda \sum_j \Gamma_{j,i} sgn(Y_{j,i}) \frac{e^{-sgn(Y_{j,i})(\hat{X}_{j,:} Q (P^t{}_{i,:})^T + E_{j,i})}}{1 + e^{-sgn(Y_{j,i})(\hat{X}_{j,:} Q (P^t{}_{i,:})^T + E_{j,i})}} \hat{X}_{j,:} Q) \\ G_{i,:}^{t+1} &= G_{i,:}^t - \eta(\lambda_E G_{i,:}^t \\ &- \lambda \sum_j \Gamma_{i,j} sgn(Y_{i,j}) \frac{e^{-sgn(Y_{i,j})(\hat{X}_{i,:}(S_{j,:})^T + G_{i,:}^t (H_{j,:})^T)}}{1 + e^{-sgn(Y_{i,j})(\hat{X}_{i,:}(S_{j,:})^T + G_{i,:}^t (H_{j,:})^T)}} H_{j,:}) \end{aligned} \tag{9}$$

where η is the learning rate of SGD, and $T_{i,j} = U_i V_j^T - X_{i,j}$. Since vanilla SGD is sensitive to the learning rate, we choose ADAM as an SGD variant with a self-adaptive learning rate [14].

We assume p iterations on average for each ADAM based optimisation operation. The computational complexity per iteration in Algorithm 1 counts to $O(p(N+M)Dk_X + p|\Omega_X|k_X + pN(D+M)k_H + p(M+N)(D+k_E))$, where $|\Omega_X|$ denotes the number of the observed feature entries of X. Though the problem is globally non-convex, the monotonic decreasing of the loss function guarantees convergence to a local minimum. Our empirical study in next section shows that the classification accuracy produced from the local minimums is satisfying and stable.

We adopt the same alternating update based optimization for learning with **ColEmbed-NL**. The non-linearity of the trigonometric transformation breaks the coordinate-wise convexity of the loss function. The convergence analysis for **ColEmbed-L** doesn't not apply for **ColEmbed-NL**. Nevertheless, ADAM empirically converges fast to local minimum during training **ColEmbed-NL**. The gradients with respect to U and V differs in **ColEmbed-NL** due to the non-linear transformation, which can be defined by chain rule:

$$\begin{aligned} \nabla_U L(U, V) &= L'(\varphi(\hat{X})) \frac{\partial \varphi(\hat{X})}{\partial \hat{X}} \frac{\partial \hat{X}}{\partial U} \\ \nabla_V L(U, V) &= L'(\varphi(\hat{X})) \frac{\partial \varphi(\hat{X})}{\partial \hat{X}} \frac{\partial \hat{X}}{\partial V} \end{aligned} \tag{10}$$

5 EMPIRICAL STUDY

5.1 Experimental settings

In this study, we investigate the performances of the proposed **ColEmbed** method for both transductive and inductive multi-label classification. Six state-of-the-art methods **WELL** [32], **LEML** [31], **DirtyIMC** [6], **MC-1** [11], **CoEmbed**[12] and **BiasMC** [13] are

Table 1: Dataset summary. *density* is the fraction of the positively labeled elements in the label matrix.

Data set	# samples	# feat.	# labels	density
Yeast	2417	103	14	30.3%
Scene	2407	294	6	17.9%
Mediamill	43907	20	101	10%
NUS-Wide Subset	50000	25	81	10%
EventCat	1150	72	6	10%

used as benchmarking baselines. **WELL**, **LEML**, **CoEmbed** and **BiasMC** target at attacking directly the issue of weak labels in multi-label learning. In contrast, the rest two methods are designed to handle semi-supervised labels, assuming both positive and negative labels are observed in the given labels. In Section 5.2 and Section.5.3, we evaluate the transductive and inductive learning performances of the involved algorithms respectively.

Four public datasets, Yeast[8], Scene[1], Mediamill[25] and NUS-Wide[7] are used for evaluation. In addition, we construct a multi-label security event categorisation dataset, collected from a real-world IoT device security application. Defence-in-depth strategy advocates customers to deploy as many as possible security products to provide the most robust defence against various threats. Security telemetry data reported by different security products can provide complementary information about security events encountered on customers' devices. By studying the highly correlated security data from different products, security vendors can thus provide better coverage on potentially malicious events. In our security event dataset, we collect security telemetry data from 1,150 network appliances, each reporting a 72 dimensional feature vector. Each dimension indicates occurring frequency of a specific type of alert. Six labels are assigned to each device in the dataset, corresponding to a variety of categories of security threats. Same telemetry features can be relevant with multiple threats. For example, scanning activity and data breaching can occur simultaneously. Event categorisation on IoT devices can be used to detect suspicious attack behaviors that can potentially compromise the IoT devices to steal private data or launch DDoS attacks. Using this dataset, we evaluate to what extent the proposed method can improve the quality of security services. Table 1 summarises information of all the five datasets (the cyber security dataset is named EventCat). *density* denotes the fraction of the positive labels in the label matrix.

The full NUS-Wide dataset contains around 0.2 millions of images, and is too large for **WELL** and **DirtyIMC**, which involve expensive full SVD on the input feature matrix. We randomly sample 50,000 images to construct a subset for benchmarking algorithms within acceptable cost. Both Mediamill and NUS-Wide datasets are labeled with high imbalance. Only 4.3% and 2.3% of elements in their label matrices are positive. In the following test, we mask out increasingly larger fraction of positively labeled elements from the label matrix of each benchmark dataset, so as to measure accuracy of reconstructing weak label assignments. Given such highly imbalanced label assignment as in Mediamill and NUS-Wide, masking over 50% of positive labels can lead to insufficient positive label elements for training, thus decreases statistical stability of any classification model involved in the study. To guarantee enough positive labels, we rank the label dimensions of each dataset according to the fraction of positively labeled elements. Label dimensions with the largest fractions of positive elements are selected. The rest of them are not used in the experiments. As a result, 20 out of 101 labels for Mediamill, 25 out of 81 label dimensions for NUS-Wide are chosen respectively to improve the coverage of positive label elements to cover 10% of the label matrix.

To evaluate the performances of transductive learning, we use all data instances of each data set. 60% of the entries from the feature matrix are randomly masked to construct incomplete features. In the label matrix, we randomly sample 30%, 50% and 80% of positive labels. These selected positive labels and all negative labels are masked as unlabeled ones, in order to construct weak labels with increasingly larger label flipping ratio τ. For the test of inductive learning, 80% of the data instances in each data set are randomly selected as training set. We follow the same sampling schemes on the feature and label matrix of the training set. The left 20% of the data instances are equipped with completed features and ground truth labels. They are used only for performance evaluation. In either mode of test, we repeat the random sampling process with replacement for 10 times. Mean and standard deviation of the derived Macro-averaging AUC [26] scores are taken as the overall evaluation metric for the algorithms. For transductive learning, we measure the classification accuracy on the reconstructed label matrix. For inductive learning, the inductive model is firstly tuned on the incomplete training set. The Macro-averaging AUC scores derived on the testing instances are used to measure the accuracy of inductive learning. To tune penalty parameters with grid search on each data set, we generate incomplete features and weak labels following the random masking scheme, which is constructed independently from those used for evaluating the accuracy of transductive learning. It is worthy to note that out-of-sample extension used in cross-validation is not inherently allowed in transductive learning in Section 5.2. Instead, we employ Gabriel hold-out patterns, a.k.a. Bi-Cross-Validation [20] to guarantee no overlapping between the data samples used for parameter tuning and performance evaluation.

For the proposed approach, k_X, k_H and k_E are set to 10 for all databases. Penalty parameters are searched over the range $[1e-2, 5e-2, 1e-1, 5e-1, 1, 5, 1e+1, 5e+1, 1e+2]$. α of the proposed approach is set to 0.2, corresponding to the case where the label flipping ratio ρ equaling to 0.6. The ground truth of ρ is never unveiled during practical usage. Therefore, we choose ρ to be slightly larger than 0.5, mimicking the applications where most of positive labels are missing. To integrate random feature expansion into **ColEmbed-NL**, we set the variance of RBF kernel as 0.1 and the number of sampling rounds as 700 in all the experiments.

5.2 Empirical study of transductive learning

Besides the six baseline methods, we include inductive extension of **BiasMC**, noted as **BiasMC-I** and two variants of **LEML** in the transductive test. The first one uses the cost-sensitive binomial loss in Eq. (1) and the second one adopts the least squared loss [31]. The two variants are denoted as **LEML-B** and **LEML-S** respectively. To conduct a fair comparison, we use the low-rank matrix completion

Table 2: Mean (standard deviation) of transductive Macro-AUC of all involved algorithms on `Yeast` and `Scene` datasets

Yeast				Scene			
Algorithm	$\tau = 0.300$	$\tau = 0.500$	$\tau = 0.800$	Algorithm	$\tau = 0.300$	$\tau = 0.500$	$\tau = 0.800$
ColEmbed-NL	**0.915(6.11e-3)**	**0.856(1.05e-3)**	**0.812(1.31e-3)**	**ColEmbed-NL**	**0.931(1.04e-3)**	**0.927(3.87e-3)**	**0.918(3.60e-3)**
ColEmbed-L	**0.901(6.23e-3)**	**0.837(5.05e-3)**	**0.761(2.33e-3)**	**ColEmbed-L**	**0.905(5.21e-3)**	**0.889(8.01e-3)**	**0.873(3.40e-3)**
MC-1	0.776(2.32e-2)	0.775(2.54e-2)	0.754(6.61e-3)	**MC-1**	0.537(1.39e-2)	0.534(8.11e-3)	0.529(8.44e-3)
DirtyIMC	0.788(8.05e-3)	0.784(8.12e-3)	0.764(8.00e-3)	**DirtyIMC**	0.881(8.31e-3)	0.854(6.34e-3)	0.782(5.15e-3)
CoEmbed	0.770(2.98e-3)	0.757(1.03e-2)	0.730(1.21e-2)	**CoEmbed**	0.798(5.10e-2)	0.767(7.09e-2)	0.728(2.25e-2)
BiasMC	0.833(1.33e-2)	0.729(1.59e-2)	0.700(6.89e-3)	**BiasMC**	0.645(5.00e-2)	0.604(2.33e-2)	0.516(2.27e-2)
BiasMC-I	0.793(1.33e-2)	0.709(1.59e-2)	0.660(6.89e-3)	**BiasMC-I**	0.675(5.00e-2)	0.624(2.33e-2)	0.576(2.27e-2)
LEML-B	0.654(4.46-3)	0.646(4.11e-3)	0.631(4.96e-3)	**LEML-B**	0.514(4.11e-3)	0.517(1.89e-3)	0.533(2.27e-2)
LEML-S	0.436(9.10e-3)	0.428(9.89e-3)	0.421(5.36e-3)	**LEML-S**	0.511(4.89e-2)	0.522(2.83e-2)	0.519(3.48e-2)
WELL	0.671(8.87e-3)	0.650(1.01e-2)	0.633(1.00e-2)	**WELL**	0.618(5.09e-2)	0.574(2.35e-2)	0.482(1.15e-2)

Table 3: Mean (standard deviation) of transductive Macro-AUC of all involved algorithms on `Mediamill` and `NUS-Wide Subset` datasets

Mediamill				NUS-Wide Subset			
Algorithm	$\tau = 0.300$	$\tau = 0.500$	$\tau = 0.800$	Algorithm	$\tau = 0.300$	$\tau = 0.500$	$\tau = 0.800$
ColEmbed-NL	**0.900(7.11e-4)**	**0.893(8.12e-4)**	**0.881(1.05e-3)**	**ColEmbed-NL**	**0.740(7.25e-3)**	**0.720(6.71e-3)**	**0.703(9.62e-3)**
ColEmbed-L	**0.905(2.10e-3)**	**0.891(5.43e-3)**	**0.865(9.12e-3)**	**ColEmbed-L**	**0.714(1.37e-2)**	**0.671(1.03e-2)**	**0.643(1.70e-2)**
MC-1	0.562(6.23e-3)	0.538(5.00e-3)	0.545(8.12e-3)	**MC-1**	0.617(2.33e-2)	0.559(1.95e-2)	0.531(1.86e-2)
DirtyIMC	0.850(7.33e-3)	0.824(5.61e-3)	0.731(1.00e-2)	**DirtyIMC**	0.709(3.99e-2)	0.655(1.95e-2)	0.630(1.00e-2)
CoEmbed	0.761(7.10e-2)	0.741(5.97e-2)	0.681(5.53e-2)	**CoEmbed**	0.695(5.31e-2)	0.611(3.05e-2)	0.583(3.28e-2)
BiasMC	0.890(8.10e-3)	0.857(8.15e-3)	0.744(1.19e-2)	**BiasMC**	0.577(5.06e-2)	0.577(8.21e-2)	0.552(5.97e-2)
BiasMC-I	0.887(9.08e-3)	0.835(8.85e-3)	0.734(1.72e-2)	**BiasMC-I**	0.613(4.20e-2)	0.605(2.85e-2)	0.593(3.77e-2)
LEML-B	0.782(7.65e-3)	0.783(5.59e-3)	0.789(4.04e-3)	**LEML-B**	0.639(3.55e-3)	0.640(6.37e-3)	0.635(6.66e-3)
LEML-S	0.550(3.63e-3)	0.545(5.54e-3)	0.536(7.93e-3)	**LEML-S**	0.638(3.28e-2)	0.636(3.94e-3)	0.632(4.13e-3)
WELL	0.618(3.20e-2)	0.593(2.58e-2)	0.430(1.73e-2)	**WELL**	0.523(5.00e-2)	0.468(3.00e-2)	0.392(3.10e-2)

Table 4: Mean (standard deviation) of transductive Macro-AUC of all involved algorithms on `EventCat`

Algorithm	$\tau = 0.300$	$\tau = 0.500$	$\tau = 0.800$
ColEmbed-NL	**0.827(7.11e-3)**	**0.781(6.55e-3)**	**0.743(1.00e-2)**
ColEmbed-L	**0.793(6.00e-3)**	**0.722(1.05e-2)**	**0.725(2.31e-2)**
MC-1	0.625(2.77e-3)	0.580(1.00e-2)	0.564(1.00e-2)
DirtyIMC	0.797(1.07e-2)	0.725(7.31e-3)	0.716(1.00e-2)
CoEmbed	0.766(6.96e-2)	0.671(7.00e-2)	0.698(3.34e-2)
BiasMC	0.685(1.97e-2)	0.623(1.93e-2)	0.532(2.00e-2)
BiasMC-I	0.675(1.77e-2)	0.625(2.00e-2)	0.512(2.30e-2)
LEML-B	0.683(5.45e-2)	0.676(6.47e-2)	0.617(2.76e-2)
LEML-S	0.615(4.12e-2)	0.615(3.76e-2)	0.602(2.44e-2)
WELL	0.487(3.00e-2)	0.437(1.57e-2)	0.363(1.41e-2)

[5], noted as **MC-Convex**, to recover side information for all the baseline methods except **MC-1** and **BiasMC**.

Table 2, Table 3 and Table 4 summarise the comparison results of transductive multi-label learning. The results of the comparative study confirm the superior performances of both the linear and non-linear variant of the proposed collaborative embedding method, compared to the baseline methods involved in the study. **DirtyIMC**, **CoEmbed** and **BiasMC** perform the best in the baseline methods. Benefited from dirty data model, e.g. the L1-norm based deviation measure, **DirtyIMC** is robust against the artefacts introduced by completing the missing elements of the feature matrix. Furthermore, **CoEmbed** gains the descriptive power by explicitly enforcing a predicative constraint on the subspace representation of features and labels in the model. However, the performances of both methods deteriorate because of the bias of weak class labels in the positive-unlabeled label assignments. In addition, as shown by the results of **BiasMC-I**, simply using the imputed feature matrix as side information in **BiasMC** doesn't necessarily improve the precision of label recovery, due to the artefacts of the imputed features.

We further measure the reconstruction error of the reconstructed feature matrix on each of the datasets in the study. In the Table 5, we compare the reconstruction error of the missing feature elements using the proposed method, **MC-Convex** and **MC-1** with $\tau = 0.8$. It is worth to note that better reconstructing features is beyond the scope of this study. The purpose of involving the comparison of the reconstruction error is to verify empirically the basic assumption of the proposed method: incomplete features and labels can provide complementary information to each other, so as to better recover the missing elements of the feature matrix. The reconstruction error is measured as $\sqrt{\sum_{i,j \notin \Omega} |X_{i,j} - \hat{X}_{i,j}|^2}$. As seen in the table, **ColEmbed-L** and **ColEmbed-NL** achieve generally higher reconstruction accuracy than the other two opponents, especially on the `Mediamill` dataset. With high fraction of missing information, the proposed method produces simultaneously good feature and label reconstruction. The results illustrate the effectiveness of the design of the proposed algorithm.

Table 5: Mean (standard deviation) of feature reconstruction error over all the datasets with $\tau = 0.8$

Dataset	**ColEmbed**	**ColEmbed-NL**	MC-Convex	MC-1
Yeast	**32.971(0.153)**	**33.011(0.122)**	33.920(0.127)	38.439(0.440)
Scene	**87.282(2.77e-3)**	**87.135(1.00e-2)**	88.430(0.982)	161.500(1.00e-2)
Mediamill	**31.625(1.669)**	**31.115(1.235)**	156.767(5.733)	297.382(3.982)
NUS-Wide Subset	**138.131(6.962)**	**142.313(7.002)**	139.938(3.114)	140.221(3.342)
Event	**1762(6.183)**	**1850(5.996)**	1896.00(8.649)	3584.00(5.966)

Table 6: Mean (standard deviation) of inductive Macro-AUC of all involved algorithms on `Yeast` and `Scene` datasets

`Yeast`				`Scene`			
Algorithm	$\tau = 0.300$	$\tau = 0.500$	$\tau = 0.800$	Algorithm	$\tau = 0.300$	$\tau = 0.500$	$\tau = 0.800$
ColEmbed-NL	**0.829(7.01e-3)**	**0.805(1.00e-2)**	**0.807(5.31e-2)**	**ColEmbed-NL**	**0.931(7.04e-3)**	**0.927(1.18e-2)**	**0.913(1.60e-2)**
ColEmbed-L	**0.650(5.11e-3)**	**0.641(6.63e-3)**	**0.634(5.51e-3)**	**ColEmbed-L**	**0.904(7.04e-3)**	**0.877(1.18e-2)**	**0.873(1.60e-2)**
DirtyIMC	0.643(8.05e-3)	0.635(8.12e-3)	0.636(8.00e-3)	**DirtyIMC**	0.732(8.31e-3)	0.712(6.34e-3)	0.670(5.15e-3)
DirtyIMC-RFE	0.775(8.05e-3)	0.764(8.12e-3)	0.755(8.00e-3)	**DirtyIMC-RFE**	0.588(8.31e-3)	0.582(6.34e-3)	0.600(5.15e-3)
CoEmbed	0.561(2.50e-2)	0.561(1.75-2)	0.5320(1.70e-2)	**CoEmbed**	0.507(1.17e-2)	0.500(7.04e-3)	0.500(1.23e-2)
CoEmbed-RFE	0.557(8.39e-3)	0.542(1.63e-2)	0.514(3.87e-2)	**CoEmbed-RFE**	0.503(1.10e-2)	0.510(1.05e-2)	0.518(1.28e-2)
LEML-B	0.635(1.33e-2)	0.640(1.59e-2)	0.620(6.89e-3)	**LEML-B**	0.523(5.00e-2)	0.521(2.33e-2)	0.526(2.27e-2)
LEML-S	0.638(1.33e-2)	0.631(1.59e-2)	0.631(6.89e-3)	**LEML-S**	0.561(5.00e-2)	0.550(2.33e-2)	0.554(2.27e-2)
LEML-B-RFE	0.776(1.33e-2)	0.775(1.59e-2)	0.776(6.89e-3)	**LEML-B-RFE**	0.487(5.00e-2)	0.477(2.33e-2)	0.480(2.27e-2)
LEML-S-RFE	0.696(1.33e-2)	0.681(1.59e-2)	0.706(6.89e-3)	**LEML-S-RFE**	0.606(5.00e-2)	0.551(2.33e-2)	0.493(2.27e-2)
BiasMC-I	0.661(1.33e-2)	0.638(1.59e-2)	0.634(6.89e-3)	**BiasMC-I**	0.820(5.00e-2)	0.813(2.33e-2)	0.793(2.27e-2)
BiasMC-I-RFE	0.772(1.33e-2)	0.764(1.59e-2)	0.771(6.89e-3)	**BiasMC-I-RFE**	0.545(5.00e-2)	0.535(2.33e-2)	0.535(2.27e-2)

Table 7: Mean (standard deviation) of inductive Macro-AUC of all involved algorithms on `Mediamill` and `NUS-Wide Subset` datasets

`Mediamill`				`NUS-Wide Subset`			
Algorithm	$\tau = 0.300$	$\tau = 0.500$	$\tau = 0.800$	Algorithm	$\tau = 0.300$	$\tau = 0.500$	$\tau = 0.800$
ColEmbed-NL	**0.889(8.24e-3)**	**0.886(1.67e-3)**	**0.886(8.18e-3)**	**ColEmbed-NL**	**0.723(7.25e-3)**	**0.706(6.71e-3)**	**0.700(9.62e-3)**
ColEmbed-L	**0.872(8.28e-3)**	**0.875(9.50e-3)**	**0.873(1.25e-2)**	**ColEmbed-L**	**0.712(1.37e-2)**	**0.702(1.30e-2)**	**0.661(1.20e-2)**
DirtyIMC	0.785(5.16e-3)	0.780(8.64e-3)	0.782(5.39e-3)	**DirtyIMC**	0.681(4.11e-3)	0.682(9.12e-3)	0.654(5.44e-3)
DirtyIMC-RFE	0.787(6.43e-3)	0.787(9.42e-3)	0.788(6.00e-3)	**DirtyIMC-RFE**	0.539(8.87e-3)	0.541(1.12e-2)	0.532(3.14e-3)
CoEmbed	0.794(1.64e-2)	0.795(9.85-3)	0.668(1.35e-2)	**CoEmbed**	0.526(1.58e-2)	0.511(1.58e-2)	0.513(2.24e-2)
CoEmbed-RFE	0.690(8.66e-3)	0.650(1.15e-2)	0.545(6.70e-2)	**CoEmbed-RFE**	0.560(1.42e-2)	0.535(5.18e-2)	0.530(1.90e-2)
LEML-B	0.793(7.13e-3)	0.789(6.60e-3)	0.780(6.83e-3)	**LEML-B**	0.703(3.55e-3)	0.701(2.96e-3)	0.683(7.27e-3)
LEML-S	0.576(5.49e-3)	0.571(6.91e-2)	0.565(9.35e-3)	**LEML-S**	0.692(5.00e-2)	0.684(2.33e-2)	0.681(2.27e-2)
LEML-B-RFE	0.788(1.35e-2)	0.788(1.59e-2)	0.790(2.32e-2)	**LEML-B-RFE**	0.618(4.56e-3)	0.615(3.67e-3)	0.616(3.52e-3)
LEML-S-RFE	0.570(1.98e-2)	0.582(1.39e-2)	0.544(1.53e-2)	**LEML-S-RFE**	0.564(7.11e-3)	0.562(8.38e-3)	0.564(2.27e-3)
BiasMC-I	0.805(1.46e-2)	0.800(1.19e-2)	0.785(7.11e-3)	**BiasMC-I**	0.696(3.10e-3)	0.693(2.05e-3)	0.683(9.30e-3)
BiasMC-I-RFE	0.818(2.03e-2)	0.798(1.02e-2)	0.792(8.32e-3)	**BiasMC-I-RFE**	0.532(3.73e-2)	0.524(9.97e-3)	0.515(1.05e-2)

5.3 Empirical study of inductive learning

Except **BiasMC** and **MC-1**, all the other baseline methods are used for inductive learning. Similarly, we use **MC-Convex** to reconstruct the feature matrix as the recovered side information. Furthermore, we extend the baseline method by applying RFE on the recovered features. The non-linear transformation of the features is used as side information in the four benchmarks. By introducing non-linearity into these methods, we aim at investigating the impact of the non-linear feature extraction in the challenging multi-label learning task. The non-linear extensions of the benchmark methods are noted as **LEML-B-RFE**, **LEML-S-RFE**, **CoEmbed-RFE**, **DirtyIMC-RFE** and **BiasMC-I-RFE**.

The detailed comparison is shown in Table 6, Table 7 and Table 8. The results show that **ColEmbed-L** and **ColEmbed-NL** achieve distinctively better inductive classification on the testing samples compared with the other inductive learning methods. The best baseline inductive methods, such as **DirtyIMC** and **BiasMC-I** and **LEML**, address either incomplete feature or weak labels. Similarly as in the transductive test, they suffer when both missing feature observations and weak labels are witnessed. By comparison, our method combines both the robustness and the capability of handling positive-unlabeled weak labels into the jointly optimisation framework, thus produce stable and accurate classification.

As we can find, random feature expansion brings more descriptive power to the proposed **ColEmbed** method, generating the best

Table 8: Mean (standard deviation) of inductive Macro-AUC of all involved algorithms on EventCat

Algorithm	$\tau = 0.300$	$\tau = 0.500$	$\tau = 0.800$
ColEmbed-NL	**0.841(1.04e-2)**	**0.762(9.05e-3)**	**0.705(2.04e-2)**
ColEmbed-L	**0.725(8.25e-3)**	**0.682(1.15e-2)**	**0.664(1.12e-2)**
DirtyIMC	0.680(1.07e-2)	0.657(1.31e-2)	0.625(1.00e-2)
DirtyIMC-RFE	0.720(3.63e-2)	0.691(4.37e-2)	0.658(1.10e-2)
CoEmbed	0.631(2.17e-2)	0.622(7.00e-3)	0.557(3.48e-2)
CoEmbed-RFE	0.553(6.87e-2)	0.549(4.61e-2)	0.519(1.24e-2)
LEML-B	0.685(1.28e-2)	0.644(1.61e-2)	0.593(1.86e-2)
LEML-S	0.547(1.15e-2)	0.535(9.23e-3)	0.516(8.66e-2)
LEML-B-RFE	0.537(1.25e-2)	0.521(8.23e-3)	0.525(1.32e-2)
LEML-S-RFE	0.580(8.61e-3)	0.554(7.82e-3)	0.539(7.25e-3)
BiasMC-I	0.665(8.97e-3)	0.623(7.00e-3)	0.612(6.16e-3)
BiasMC-I-RFE	0.691(1.97e-2)	0.677(1.93e-2)	0.618(2.10e-2)

accuracy over all the datasets. In contrast, the non-linear feature extraction does not necessarily improve the other benchmark methods. Especially on Scene and Nus-Wide Subset, introducing the non-linear features per contra decreases inductive classification performances of **DirtyIMC** and **LEML-S**.

The major cause is the co-occurrence of the artifacts in the recovered features and the weak labels. Both issues, as a whole, inject unpredictable fluctuation to the non-linear classification model. In our method, tuning of the non-linear inductive model is coupled with learning of the predictive relation between partially observed features and labels. The jointly learning process suppresses the negative impact thus produces better approximation to the underlying class distribution.

5.4 Run Time Test

We measure the run time cost of the proposed **ColEmbed-L** and **ColEmbed-NL** on the data sets of increasingly larger size. They are implemented using *NumPy* and *Theano* packages and run on a MacBook Pro laptop with Intel Core i7 2.5GHz CPU and 16GB DDR3 RAM. Table.9 shows the measured run time on the five data sets used for benchmarking. In addition, to observe the variation tendency of the computational cost of the proposed method. we randomly sample the full *NUS-Wide* dataset to generate subsets with increasingly larger size scaled by 5, containing 1000, 5000, 25000 and 125000 data instances. On each subset, both algorithms run for 10 times. The average running time measured in seconds is reported in Fig.1 with logarithmic scale in both axes. In the running time test, we follow the setting of transdutive learning where all data instances are used in model training, with τ fixed to 0.8. According to Table.9, the run time cost fluctuates over different datasets. It is mainly due to the varied distributional characteristics of different datasets, which change difficulty of optimization. From Fig.1, we can find clearly that the computational cost of the proposed method scales up at a linear rate.

6 CONCLUSION

We attack the challenging issue of multi-label learning with highly incomplete data by collaboratively learning missing features and labels. Alongside with jointly feature and label recovery, a prediction function is learnt to fit the underlying association between feature representation and label assignments. Thanks to the flexible design, the proposed collaborative embedding method can conduct both transductive and inductive learning simultaneously. Extensive experimental study illustrate consistently good applicability of the proposed method for practical multi-label learning tasks. In our future plan, we plan to integrate more powerful prediction model, like neural nets, to improve the performances in a further step.

Table 9: Averaged run time (seconds) on the five datasets

Data set	**ColEmbed-L**	**ColEmbed-NL**
Yeast	3.255	26.225
Scene	6.954	60.133
Mediamill	605.429	4591.193
NUS-Wide Subset	34.984	405.233
EventCat	10.435	66.076

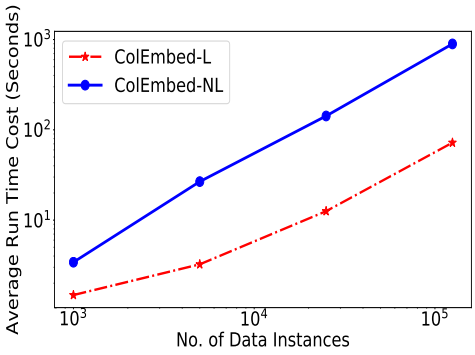

Figure 1: Scalability of ColEmbed-L and ColEmbed-NL measured in logarithmic scale

ACKNOWLEDGMENTS

This work is partially supported by King Abdullah University of Science and Technology (KAUST).

REFERENCES

[1] Matthew R. Boutell, Jiebo Luo, Xipeng Shen, and Christopher M. Brown. Learning multi-label scene classification. *Pattern Recognition*, 37:1757–1771, 2004.
[2] Serhat Selcuk Bucak, Rong Jin, and Anil K Jain. Multi-label learning with incomplete class assignments. In *CVPR*, pages 2801–2808, June 2011.
[3] Ricardo Cabral, Fernando De la Torre, Joao Paulo Costeira, and Alexandre Bernardino. Matrix completion for weakly-supervised multi-label image classification. *TPAMI*, 37(1):121–135, 2015.
[4] Borja Calvo, Pedro Larranage, and Jose A.Lozano. Feature subset selection from positive and unlabelled examples. *Pattern Recognition Letters*, 30:1027–1036, 2009.
[5] Emmanuel J Candes and Yaniv Plan. Matrix completion with noise. *Proceedings of the IEEE*, 98(6):925–936, June 2010.
[6] Kai-Yang Chiang, Cho-Jui Hsieh, and Inderjit S. Dhillon. Matrix completion with noisy side information. In *NIPS*, pages 3447–3455, 2015.
[7] Tat-Seng Chua, Jinhui Tang, Richang Hong, Haojie Li, Zhiping Luo, and Yantao Zheng. NUS-WIDE: A real-world web image database from national university of singapore. In *Proceedings of the ACM International Conference on Image and Video Retrieval*, pages 48:1–48:9, 2009.
[8] André Elisseeff and Jason Weston. A kernel method for multi-labelled classification. In *NIPS*, pages 681–687, 2001.
[9] Charles Elkan and Keith Noto. Learning classifiers from only positive and unlabeled data. In *SIGKDD*, pages 213–220, 2008.

[10] Gabriel Pui Cheong Fung, J. X. Yu, Hongjun Lu, and P. S. Yu. Text classification without negative examples revisit. *TKDE*, 18(1):6–20, 2006.
[11] Andrew B. Goldberg, Xiaojin Zhu, Benjamin Recht, Jun-Ming Xu, and Robert Nowak. Transduction with matrix completion: Three birds with one stone. In *NIPS*, pages 757–765, 2010.
[12] Yuhong Guo. Convex co-embedding for matrix completion with predictive side information. In *AAAI*, pages 1955–1961, 2017.
[13] Cho-Jui Hsieh, Nagarajan Natarajan, and Inderjit S.Dhillon. PU learning for matrix completion. In *ICML*, pages 663–672, 2015.
[14] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2014.
[15] Xiaoli Li and Bing Liu. Learning to classify texts using positive and unlabeled data. In *IJCAI*, pages 587–592, 2003.
[16] Zijia Lin, Guiguang Ding, Mingqing Hu, Jianmin Wang, and Xiaojun Ye. Image tag completion via image-specific and tag-specific linear sparse reconstructions. In *CVPR*, pages 1618–1625, 2013.
[17] Dong Liu, Xian-Sheng Hua, Meng Wang, and Hong-Jiang Zhang. Image retagging. In *Proceedings of the 18th ACM International Conference on Multimedia*, pages 491–500, 2010.
[18] Farzaneh Mirzazadeh, Yuhong Guo, and Dale Schuurmans. Convex co-embedding. In *AAAI*, pages 1989–1996, 2014.
[19] Nagarajan Natarajan, Inderjit S Dhillon, Pradeep K Ravikumar, and Ambuj Tewari. Learning with noisy labels. In *NIPS*, pages 1196–1204, 2013.
[20] Art B Owen and Patrick O Perry. Bi-cross-validation of the SVD and the nonnegative matrix factorization. *The Annals of Applied Statistics*, 3(2):564–594, 2009.
[21] Marthinus C. du Plessis, Gang Niu, and Masashi Sugiyama. Analysis of learning from positive and unlabeled data. In *NIPS*, pages 703–711, 2014.
[22] Ali Rahimi and Ben Recht. Random features for large-scale kernel machines. In *NIPS*, 2007.
[23] Sundararajan Sellamanickam, Priyanka Garg, and Sathiya Keerthi Selvaraj. Pairwise ranking based approach to learning with positive and unlabelled examples. In *CIKM*, pages 663–672, 2011.
[24] Si Si, Kai-Yang Chiang, Cho-Jui Hsieh, Nikhil Rao, and Inderjit Dhillon. Goal-directed inductive matrix completion. In *KDD*, 2016.
[25] Cees G. M. Snoek, Marcel Worring, Jan C. van Gemert, Jan-Mark Geusebroek, and Arnold W. M. Smeulders. The challenge problem for automated detection of 101 semantic concepts in multimedia. In *Proceedings of the 14th ACM International Conference on Multimedia*, pages 421–430, 2006.
[26] Grigorios Tsoumakas, Ioannis Katakis, and Ioannis P Vlahavas. Random k-labelsets for multilabel classification. *TKDE*, 23(7):1079–1089, 2011.
[27] Changhu Wang, Shuicheng Yan, Lei Zhang, and Hongjiang Zhang. Multi-label sparse coding for automatic image annotation. In *CVPR*, pages 1643–1650, 2009.
[28] Lei Wu, Rong Jin, and Anil K Jain. Tag completion for image retrieval. *TPAMI*, 35(3):716–727, 2013.
[29] Miao Xu, Rong Jin, and Zhi-Hua Zhou. Speedup matrix completion with side information: Application to multi-label learning. In *NIPS*, pages 2301–2309, 2013.
[30] Hsiang-Fu Yu, Prateek Jain, Purushottam Kar, and Inderjit S. Dhillon. Large-scale multi-label learning with missing labels. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ICML'14, pages I–593–I–601. JMLR.org, 2014.
[31] Hsiang-Fu Yu, Prateek Jain, Purushottam Kar, and Inderjit S.Dhillon. Large-scale multi-label learning with missing labels. In *ICML*, 2014.
[32] Yin Zhang Yu-Yin Sun and Zhi-Hua Zhou. Multi-label learning with weak label. In *AAAI*, pages 593–598, 2010.
[33] Feipeng Zhao and Yuhong Guo. Semi-supervised multi-label learning with incomplete labels. In *IJCAI*, pages 4062–4068, 2015.
[34] Guangyu Zhu, Shuicheng Yan, and Yi Ma. Image tag refinement towards low-rank, content-tag prior and error sparsity. In *Proceedings of the 18th ACM International Conference on Multimedia*, pages 461–470, 2010.