Pattern Recognition 74 (2018) 488–502

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

Multi-label feature selection with missing labels

Pengfei Zhu*, Qian Xu, Qinghua Hu, Changqing Zhang, Hong Zhao

School of Computer Science and Technology, Tianjin Univerity, Tianjin 300350, China

ARTICLE INFO

Article history:
Received 21 March 2017
Revised 26 June 2017
Accepted 24 September 2017
Available online 25 September 2017

Keywords:
Feature selection
Multi-label learning
Missing labels

ABSTRACT

The consistently increasing of the feature dimension brings about great time complexity and storage burden for multi-label learning. Numerous multi-label feature selection techniques are developed to alleviate the effect of high-dimensionality. The existing multi-label feature selection algorithms assume that the labels of the training data are complete. However, this assumption does not always hold true for labeling data is costly and there is ambiguity among classes. Hence, in real-world applications, the data available usually have an incomplete set of labels. In this paper, we present a novel multi-label feature selection model under the circumstance of missing labels. With the proposed algorithm, the most discriminative features are selected and missing labels are recovered simultaneously. To remove the irrelevant and noisy features, the effective $l_{2,p}$-norm $(0 < p \leq 1)$ regularization item is imposed on the feature selection matrix. To solve the optimization problem, we developed an iterative reweighted least squares (IRLS) algorithm with guaranteed convergence. Experimental results on benchmark datasets show that the proposed method outperforms the state-of-the-art multi-label feature selection algorithms.

1. Introduction

Multi-label learning has been widely applied to various real-world tasks, e.g., text categorization [1], functional genomics [2] and image classification [3]. In functional genomics, each gene can be associated with several functional classes, e.g. *metabolism* and *transcription*. In image classification, each image can be tagged with various labels to convey multiple concepts. For instance, the left sub-figure of Fig. 1 can be labelled by *skating, crowd, winter*, and *bridge*. Under these circumstances, one label per instance is out of capability to express the underlying complex semantics. Hence, multi-label learning is developed by assigning multiple labels to one instance simultaneously [4].

Generally, the dimension of multi-label text, gene and image data can be very high for their rich semantic contents [5]. As a result, multi-label learning suffers from high dimensionality of the feature space [6,7]. The curse of dimensionality leads to the exponential growth of model parameters, which greatly degrade the generalization ability of the multi-label learning machines. In fact, only a small subset of features is informative and many features can be irrelevant and redundant. Therefore, multi-label feature selection is proposed to select the most representative features and alleviate the effect of high-dimensionality for multi-label data. Earlier works such as [3,8] are two-stage methods. Multi-label data are first transformed to single-label data, and then traditional single-label feature selection approaches are applied. Afterwards, researchers proposed to extend the existed algorithms to multi-label versions. Yu et al. extended latent semantic indexing (LSI) to a multi-label edition which considers the input feature space and output label space simultaneously [9]. Zhang and Zhou obtained a lower dimensional feature space by maximizing the dependence between the original feature description and the class labels [10]. Wang et al. extended the classical Linear Discriminant Analysis (LDA) to the multi-label formulation [11].

Traditional supervised multi-label feature selection approaches hold the assumption that the label matrix of training samples is complete. That is, the whole labels of training samples have been given. Unfortunately, the labels available are usually incomplete for the great difficulty of getting the whole proper labels. Firstly, the cost of getting amounts of labeled data is extremely expensive, especially for large-scale tasks [12]. Moreover, the great ambiguity among classes also prevents us from labelling completely and correctly. As shown in Fig. 1, the left sub-figure is annotated with *skating, crowd, winter*, or *bridge* while other labels, e.g., *park, children, tree, New York*, perhaps are missing. In the right sub-figure, it is difficult to label the emotion or expression of a face image because of the significant class ambiguity. Hence, the full label matrix assumption is unpractical and does not hold for some real-world applications.

Under the missing circumstance, approaches which directly model the original complete label matrix cannot accurately cap-

* Corresponding author.
E-mail address: zhupengfei@tju.edu.cn (P. Zhu).

https://doi.org/10.1016/j.patcog.2017.09.036
0031-3203/© 2017 Elsevier Ltd. All rights reserved.

Table 1
Descriptions of the benchmark datasets.

Dataset	Sample	Feature	Class	Train	Test	Domain
Artificial	5000	462	26	2000	3000	Text
Bookmarks	5000	2150	208	2000	3000	Text
Birds	645	260	19	322	323	Audio
Reference	5000	793	33	2000	3000	Text
Social	5000	1047	39	2000	3000	Text
Yeast	2417	103	14	1499	918	Biology

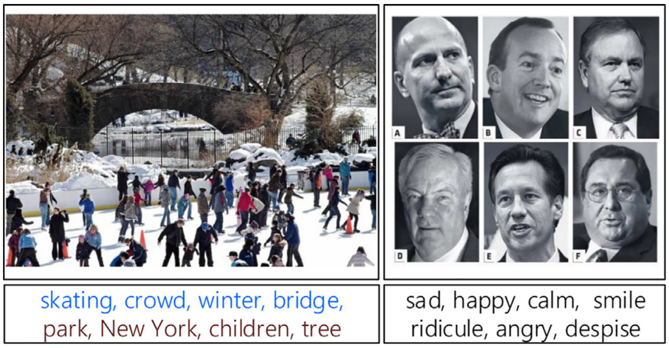

A
B
C
D
E
F
skating, crowd, winter, bridge,
park, New York, children, tree
sad, happy, calm, smile
ridicule, angry, despise

Fig. 1. Two typical examples of multi-label learning with missing labels, i.e., *image annotation* and *emotion & expression recognition*.

ture the relations among labels anymore. Following Wu et al. [13], we refer to this problem as *multi-label learning with missing labels* (MLML). To effectively deal with MLML problem, we proposed a novel algorithm called *multi-label feature selection with missing labels* (MLMLFS). Specifically, a linear regression model is used to recover the missing labels and the group lasso constraint is imposed on the projection matrix to select features at the same time. Additionally, instance-level dependency is incorporated to require that similar instances have similar labels. This work is an extended version of our conference paper [14]. We further introduced the label dependency and reformulated the model of multi-label feature selection with missing labels. The main contributions of this paper are summarized as below.

- To the best of our knowledge, this is the first attempt to conduct feature selection for multi-label classification with missing labels.
- An embedded feature selection method is proposed with which feature selection can be conducted during the process of label recovery.
- The effective $l_{2,p}$-norm $(0 < p \leq 1)$ regularization is imposed on the feature selection matrix to select the most discriminative features and remove noisy ones at the same time.
- Label dependency is incorporated into the model to require that similar samples in the feature space stay close in the recovered label space, which helps exploit label correlations.

The rest of this paper is organized as follows. Section 2 reviews the related works. In Section 3, we give the basic notations and introduce the objective. In Section 4, we propose our robust multi-label feature selection model to deal with missing labels. Section 5 conducts experiments and Section 6 concludes.

Table 2
Evaluation under 0% missing labels on benchmark datasets.

(a) Hamming loss

Data	MDDM	PMU	SFUS	MDMR	MLMLFS $(p = 0.4)$	MLMLFS $(p = 0.6)$	MLMLFS $(p = 0.8)$	MLMLFS $(p = 1)$
Artificial	0.0620	0.0603	0.0605	0.0587	**0.0566**	**0.0566**	0.0568	0.0569
Birds	0.0591	0.0599	0.0515	0.0571	**0.0478**	**0.0478**	**0.0478**	**0.0478**
Bookmarks	0.0356	0.0328	0.0306	0.0319	0.0294	0.0288	0.0275	**0.0271**
Reference	0.0310	0.0295	0.0299	0.0295	**0.0276**	**0.0276**	0.0279	0.0279
Social	0.0263	0.0247	0.0247	0.0244	**0.0221**	0.0222	0.0222	**0.0221**
Yeast	0.1979	0.1993	0.1986	0.1987	**0.1974**	**0.1974**	**0.1974**	**0.1974**

(b) One error

Data	MDDM	PMU	SFUS	MDMR	MLMLFS $(p = 0.4)$	MLMLFS $(p = 0.6)$	MLMLFS MLMLFS $(p = 0.8)$	$(p = 1)$
Artificial	0.6650	0.6347	0.6177	0.6060	0.5497	0.5537	0.5517	**0.5440**
Birds	0.4303	0.4582	0.3715	0.3994	**0.3096**	0.3127	**0.3096**	0.3282
Bookmarks	0.5430	0.5322	0.2160	0.5217	0.1717	0.1617	0.1517	**0.1287**
Reference	0.5063	0.4867	0.4690	0.4870	0.4280	**0.4250**	0.4290	0.4323
Social	0.3993	0.3820	0.3803	0.3753	0.3220	0.3200	0.3217	**0.3170**
Yeast	0.2334	0.2225	0.2345	0.2334	**0.2181**	0.2246	0.2246	0.2246

(c) Average precision

Data	MDDM	PMU	SFUS	MDMR	MLMLFS $(p = 0.4)$	MLMLFS $(p = 0.6)$	MLMLFS $(p = 0.8)$	MLMLFS $(p = 1)$
Artificial	0.4881	0.5019	0.5111	0.5182	**0.5577**	0.5561	0.5556	0.5569
Birds	0.6658	0.6568	0.6904	0.6915	0.7253	**0.7272**	0.7185	0.7176
Bookmarks	0.2987	0.3031	0.4266	0.4132	0.4737	0.4882	0.5108	**0.5343**
Reference	0.5929	0.6142	0.6288	0.6157	0.6586	**0.6592**	0.6554	0.6495
Social	0.6895	0.7052	0.7056	0.7108	0.7424	0.7426	0.7453	**0.7456**
Yeast	0.7559	**0.7591**	0.7579	0.7561	0.7587	0.7589	0.7589	0.7589

(d) Macro-F

Data	MDDM	PMU	SFUS	MDMR	MLMLFS $(p = 0.4)$	MLMLFS $(p = 0.6)$	MLMLFS $(p = 0.8)$	MLMLFS $(p = 1)$
Artificial	0.0591	0.1214	0.1374	0.1669	0.2289	0.2428	0.2504	**0.2515**
Birds	0.4771	0.4373	0.5167	0.5105	0.5494	**0.5505**	0.5494	0.5494
Bookmarks	0.1715	0.2011	0.3386	0.3110	0.3941	0.4057	0.4414	**0.4630**
Reference	0.2827	0.2827	0.3504	0.2783	0.4345	**0.4729**	0.4479	0.4479
Social	0.4219	0.5336	0.4509	0.5121	0.5512	0.5548	**0.5569**	0.5566
Yeast	0.6131	0.6141	0.6168	0.6112	**0.6187**	**0.6187**	**0.6187**	**0.6187**

Table 3
Evaluation under 25% missing labels on benchmark datasets.

(a) Hamming loss

Data	MDDM	PMU	SFUS	MDMR	MLMLFS ($p = 0.4$)	MLMLFS ($p = 0.6$)	MLMLFS ($p = 0.8$)	MLMLFS ($p = 1$)
Artificial	0.0627	0.0615	0.0603	0.0595	0.0570	0.0565	**0.0563**	0.0565
Birds	0.0628	0.0635	0.0531	0.0551	0.0498	0.0489	**0.0481**	0.0498
Bookmarks	0.0359	0.0347	0.0331	0.0401	0.0312	0.0273	**0.0269**	**0.0269**
Reference	0.0313	0.0301	0.0294	0.0295	0.0277	**0.0276**	0.0278	**0.0276**
Social	0.0271	0.0263	0.0255	0.0253	0.0224	0.0220	0.0220	**0.0218**
Yeast	0.2029	0.2030	0.2004	0.2022	0.1986	0.1988	**0.1981**	0.1982

(b) One error

Data	MDDM	PMU	SFUS	MDMR	MLMLFS ($p = 0.4$)	MLMLFS ($p = 0.6$)	MLMLFS ($p = 0.8$)	MLMLFS ($p = 1$)
Artificial	0.6753	0.6413	0.6313	0.6157	0.5567	0.5533	0.5537	**0.5523**
Birds	0.4396	0.4768	0.3653	0.4056	0.3189	**0.3096**	0.3127	0.3127
Bookmarks	0.5650	0.5536	0.2827	0.5590	0.2107	0.1437	0.1317	**0.1297**
Reference	0.5173	0.5030	0.4650	0.4833	0.4250	**0.4230**	0.4277	0.4280
Social	0.4260	0.4047	0.3767	0.3760	0.3280	0.3297	**0.3227**	0.3240
Yeast	0.2497	0.2399	0.2432	0.2323	**0.2236**	0.2312	0.2290	0.2290

(c) Average precision

Data	MDDM	PMU	SFUS	MDMR	MLMLFS ($p = 0.4$)	MLMLFS ($p = 0.6$)	MLMLFS ($p = 0.8$)	MLMLFS ($p = 1$)
Artificial	0.4748	0.4967	0.5031	0.5067	0.5532	0.5535	0.5532	**0.5552**
Birds	0.6571	0.6584	0.6949	0.6751	0.7179	**0.7277**	0.7262	0.7265
Bookmarks	0.2889	0.2922	0.3807	0.3709	0.4242	0.5136	0.5300	**0.5325**
Reference	0.5827	0.6039	0.6277	0.6161	0.6551	**0.6561**	0.6512	0.6534
Social	0.6765	0.6904	0.7066	0.7030	0.7381	0.7391	**0.7409**	0.7403
Yeast	0.7486	0.7501	0.7502	0.7534	0.7561	**0.7563**	0.7555	0.7558

(d) Macro-F

Data	MDDM	PMU	SFUS	MDMR	MLMLFS ($p = 0.4$)	MLMLFS ($p = 0.6$)	MLMLFS ($p = 0.8$)	MLMLFS ($p = 1$)
Artificial	0.0435	0.0742	0.1104	0.1489	0.2286	0.2313	**0.2350**	0.2243
Birds	0.4723	0.4124	0.4779	0.4636	0.5301	**0.5388**	0.5367	0.5275
Bookmarks	0.1601	0.1879	0.2545	0.3001	0.3112	0.4430	**0.4608**	0.4578
Reference	0.2942	0.2668	0.3262	0.2933	0.4439	0.4439	0.4460	**0.4503**
Social	0.4421	0.4327	0.4612	0.5049	0.5608	0.5593	**0.5630**	0.5628
Yeast	0.6048	0.6158	0.6131	0.6129	0.6224	0.6217	0.6228	**0.6230**

2. Related work

2.1. Multi-label learning

Multi-label learning deals with instances belonging to several semantic concepts simultaneously and has been successfully applied to diverse real-world tasks. The goal of multi-label classification is to predict the proper labels of unseen instances from instances with known labels [4]. Generally, the approaches proposed to solve multi-label tasks can be classified into two categories: binary approaches and ranking approaches. Binary approaches try to decompose the multi-label problem into multiple independent binary ones. As for each classification task, instances which are associated with the label are seen as positive samples while others are seen as negative samples [3,15,16]. The binary approaches can be seen as *first-order strategies*, which track the correlations among labels with the advantages of conceptual simplicity and high efficiency. Actually, some labels are usually interdependent and correlated [4,16]. Binary approaches ignore the inherent correlations among labels. Moreover, binary methods restrict the expressive power of such a system to some extent. Another weakness lies in that when the number of labels is large, the number of binary classifiers would be much larger. This can lead to severe consequences such as sparse training samples and class imbalance. Hence, binary approaches are hard to be applied to large-scale multi-label problems. Ranking approaches transform the label prediction of unseen samples into the ranking of labels [1,17–19]. These methods encourage the more approximate labels of the sample to rank before other labels. Schapire and Singer proposed to utilize a family of improved boosting algorithms to solve the problem of text classification [1]. Elisseeff and Weston introduced a kernel method which encourages the potential labels of the instance to appear at the top of ranking list [17]. It defines a specific cost function which minimizes the ranking loss while maximizes the margin. These approaches are computationally expensive when the number of classes is large. Bucak et al. adopted an efficient ranking method which maximizes classification margin and minimizes classification errors simultaneously [18]. In image annotation, an image is usually represented by a bag of local regions, and therefore image annotation is formulated as a multi-instance multi-label problem [20]. Overall, multi-label ranking alleviates the problem of imbalanced data distribution compared with binary approaches for refraining from binary decision.

2.2. Multi-label feature selection

Multi-label feature selection which aims to reduce the dimensionality of multi-label data has attracted much attention these years. With label information, the underlying correlations among labels can be captured to enhance the performance of feature selection. The existing multi-label feature selection algorithms can be generally categorized into three types, i.e., filter, wrapper and embedding approaches. Filter methods evaluate the discrimination capacity of features by defining various statistical measurements.

Table 4
Evaluation under 50% missing labels on benchmark datasets.

(a) Hamming loss

Data	MDDM	PMU	SFUS	MDMR	MLMLFS	MLMLFS	MLMLFS	MLMLFS
					($p = 0.4$)	($p = 0.6$)	($p = 0.8$)	($p = 1$)
Artificial	0.0618	0.0621	0.0622	0.0606	0.0575	0.0573	0.0573	**0.0571**
Birds	0.0639	0.0593	0.0550	0.0593	0.0529	**0.0519**	0.0528	0.0523
Bookmarks	0.0367	0.0405	0.0353	0.0496	0.0273	0.0272	**0.0271**	0.0272
Reference	0.0348	0.0318	0.0317	0.0311	0.0279	**0.0277**	**0.0277**	0.0279
Social	0.0294	0.0264	0.0261	0.0262	0.0229	0.0230	0.0229	**0.0228**
Yeast	0.2098	0.2116	0.2091	0.2104	0.1999	0.2003	**0.1995**	0.1998

(b) One error

Data	MDDM	PMU	SFUS	MDMR	MLMLFS	MLMLFS	MLMLFS	MLMLFS
					($p = 0.4$)	($p = 0.6$)	($p = 0.8$)	($p = 1$)
Artificial	0.6767	0.6657	0.6623	0.6440	0.5750	**0.5740**	0.5763	0.5773
Birds	0.4768	0.4458	0.4241	0.4458	0.3684	**0.3560**	0.3591	0.3715
Bookmarks	0.5637	0.5644	0.3780	0.5683	0.1410	0.1380	**0.1360**	0.1367
Reference	0.5267	0.5103	0.4880	0.4970	0.4467	0.4453	**0.4373**	0.4427
Social	0.4560	0.3970	0.3960	0.3933	0.3437	0.3410	0.3363	**0.3353**
Yeast	0.2486	0.2410	0.2399	0.2399	**0.2279**	**0.2279**	**0.2279**	**0.2279**

(c) Average precision

Data	MDDM	PMU	SFUS	MDMR	MLMLFS	MLMLFS	MLMLFS	MLMLFS
					($p = 0.4$)	($p = 0.6$)	($p = 0.8$)	($p = 1$)
Artificial	0.4776	0.4825	0.4845	0.4906	**0.5364**	0.5340	0.5348	0.5347
Birds	0.6246	0.6496	0.6424	0.6312	0.6857	**0.6902**	0.6849	0.6837
Bookmarks	0.2802	0.2887	0.3161	0.3219	0.5167	0.5209	**0.5215**	0.5195
Reference	0.5748	0.5972	0.6118	0.5993	0.6420	0.6440	**0.6446**	0.6415
Social	0.6541	0.6935	0.6970	0.6939	0.7257	0.7265	**0.7290**	0.7283
Yeast	0.7376	0.7423	0.7413	0.7387	0.7518	0.7519	**0.7524**	0.7523

(d) Macro-F

Data	MDDM	PMU	SFUS	MDMR	MLMLFS	MLMLFS	MLMLFS	MLMLFS
					($p = 0.4$)	($p = 0.6$)	($p = 0.8$)	($p = 1$)
Artificial	0.0539	0.0435	0.0641	0.1155	**0.2013**	0.2007	0.1902	0.1902
Birds	0.3772	0.4685	0.4756	0.4622	**0.4962**	0.4953	0.4835	0.4894
Bookmarks	0.1562	0.1701	0.2015	0.2159	0.4544	**0.4561**	0.4540	0.4517
Reference	0.2874	0.2587	0.3188	0.2631	**0.4407**	0.4324	0.4353	0.4306
Social	0.2867	0.3982	0.4266	0.5271	**0.5460**	0.5302	0.5421	0.5379
Yeast	0.5822	0.5893	0.5907	0.5835	0.6140	0.6177	0.6155	**0.6185**

After the process of evaluation, features can be ranked according to the corresponding scores which reflect their relevance. Cherman et al. proposed to use ReliefF and an adaptation of information gain to evaluate features [21]. Alalga et al. used constrained Laplacian score to measure the similarity among features [22]. Lin et al. addressed the task of multi-label feature selection from the viewpoints of neighborhood information entropy [23], and dependency maximum and redundancy minimum [5]. Wrapper methods apply some search strategies (e.g., genetic algorithm) and rely on the performance of the chosen learning machines to select a subset of features. Feature selection and evaluation stages are executed independently and iteratively in wrapper methods. Zhang et al. adopts a filter-wrapper strategy in which principal component analysis (PCA) is used firstly to reduce the dimensionality and then genetic algorithm (GA) is used to choose the most appropriate features [24]. But this method has the drawbacks such as a slow convergence speed and premature solutions. Lee et al. employed a memetic algorithm which exploits a local refinement method to enhance the performance of genetic search [25]. For embedding methods, feature selection is incorporated into the process of model construction. To remove the irrelevant and noisy features, sparse regularization is usually imposed on the feature selection matrices. Gu et al. incorporated feature selection into LaRank SVM [26]. Wang et al. developed a sparsity-based model for feature selection by capturing the shared subspace of original space for multi-label learning [11]. Cai et al. proposed a graph structured sparsity model to address label correlations and the structural information among classes, which can also apply to multi-label feature selection tasks [27]. Embedded methods usually suffer from iterative matrix inversion calculations and can be time-consuming.

2.3. Multi-label learning with missing labels

The existence of missing labels destroys the inherent label structure and obscures the original semantic concepts, and thus has an undesirable effect on the process of modelling label correlations. For multi-label learning with missing labels, the key challenge is how to handle the missing labels. According to the type of absent labels, the problem can be grouped into *positive labels missing* and *positive & negative labels missing*. In the first case, only positive labels are missing and the key challenge is how to recover the absent positive labels. Sun et al. proposed to avoid class imbalance and require that the classification boundary for each label to go across low density regions [28]. It exploits the label correlations by assuming that similar instances should have similar labels. Bucak et al. proposed a ranking based method which uses group lasso to select the missing class assignments [29]. While in the second case, *positive label* and *negative label* are used to represent that the instance is assigned to the label or not. The absence of a label indicates that we do not know whether the instance belongs to this label or not. It is evident that the two scenarios are of much difference. Only positive labels are absent for *positive labels missing*, while for *positive & negative labels missing*, both positive and negative labels can be absent simultaneously. Yu et al. learned a low-

Table 5
Evaluation under 80% missing labels on benchmark datasets.

(a) Hamming loss

Data	MDDM	PMU	SFUS	MDMR	MLMLFS	MLMLFS	MLMLFS	MLMLFS
					($p = 0.4$)	($p = 0.6$)	($p = 0.8$)	($p = 1$)
Artificial	0.0630	0.0630	0.0629	0.0628	0.0592	**0.0586**	**0.0586**	**0.0586**
Birds	0.0661	0.0568	0.0550	0.0568	0.0542	0.0540	0.0540	**0.0531**
Bookmarks	0.0385	0.0433	0.0375	0.0531	0.0324	0.0305	**0.0300**	0.0301
Reference	0.0351	0.0317	0.0336	0.0317	0.0293	0.0293	**0.0292**	0.0293
Social	0.0298	0.0296	0.0295	0.0291	0.0242	0.0241	0.0239	**0.0235**
Yeast	0.2217	0.2251	0.2224	0.2243	0.2053	0.2053	0.2053	**0.2052**

(b) One error

Data	MDDM	PMU	SFUS	MDMR	MLMLFS	MLMLFS	MLMLFS	MLMLFS
					($p = 0.4$)	($p = 0.6$)	($p = 0.8$)	($p = 1$)
Artificial	0.7347	0.7130	0.7013	0.6960	0.6160	**0.5930**	0.6083	0.6093
Birds	0.5046	0.4551	0.4644	0.4551	0.4087	0.3963	0.3963	**0.3839**
Bookmarks	0.6090	0.5890	0.4290	0.5764	0.2063	0.1740	**0.1700**	0.1757
Reference	0.5220	0.5210	0.5093	0.5200	0.4733	**0.4727**	0.4743	0.4747
Social	0.4607	0.4470	0.4583	0.4327	0.3697	0.3650	0.3603	**0.3593**
Yeast	0.2486	0.2465	0.2497	0.2454	**0.2388**	0.2410	0.2410	0.2410

(c) Average precision

Data	MDDM	PMU	SFUS	MDMR	MLMLFS	MLMLFS	MLMLFS	MLMLFS
					($p = 0.4$)	($p = 0.6$)	($p = 0.8$)	($p = 1$)
Artificial	0.4410	0.4454	0.4522	0.4531	0.5100	**0.5213**	0.5118	0.5080
Birds	0.6077	0.6087	0.6191	0.6130	0.6451	0.6599	0.6628	**0.6658**
Bookmarks	0.2556	0.2488	0.2719	0.2630	0.4117	0.4466	**0.4554**	0.4521
Reference	0.5728	0.5770	0.5871	0.5722	0.6135	0.6149	**0.6164**	0.6135
Social	0.6535	0.6624	0.6568	0.6691	0.7068	0.7114	0.7113	**0.7123**
Yeast	0.7228	0.7192	0.7244	0.7209	0.7440	**0.7443**	0.7433	**0.7443**

(d) Macro-F

Data	MDDM	PMU	SFUS	MDMR	MLMLFS	MLMLFS	MLMLFS	MLMLFS
					($p = 0.4$)	($p = 0.6$)	($p = 0.8$)	($p = 1$)
Artificial	0.0165	0.0292	0.0430	0.0435	0.1598	**0.1805**	0.1623	0.1572
Birds	0.4220	0.4396	0.4303	0.4396	0.4752	0.4748	0.4787	**0.4966**
Bookmarks	0.1237	0.1453	0.1598	0.1790	0.3227	0.3736	**0.3852**	0.3782
Reference	0.2908	0.2092	0.2839	0.2387	**0.3987**	0.3787	0.3518	0.3541
Social	0.2660	0.3126	0.2118	0.3341	0.3987	0.4034	**0.4209**	0.4196
Yeast	0.5352	0.5408	0.5416	0.5448	0.6082	0.6082	**0.6139**	0.6094

rank matrix to capture the correlations among labels, which avoids over-fitting and is computationally efficient [30]. Chen et al. proposed to learn two classifiers from both features and incomplete tags to predicted tags [31]. Xu et al. adopted one-to-all reconstruction incorporates with a low rank structure to exploit the correlations among labels [32]. A supplementary label matrix based on label dependency propagation is to address high-order correlations among labels for incomplete label matrices. Wu et al. explicitly defined the missing labels with a zero element for unsigned labels. They utilized both label consistence and label smoothness assumptions to reconstruct the incomplete label matrix [13,33]. Specifically, Wu et al. [33] consider missing labels and class imbalance simultaneously.

3. Problem statement

Here, we give the basic notations that we used in this paper. $\mathbf{X} = \{\mathbf{x}_1, \cdots, \mathbf{x}_n\}$ is the data matrix that consists of n samples. For each sample, $\mathbf{x}_i \in \mathbb{R}^d$ and d is the feature dimension. $\mathbf{Y} = \{\mathbf{y}_1, \cdots, \mathbf{y}_n\}$ is the label matrix. For each sample, $\mathbf{y}_i \in \mathbb{R}^c$ is the label vector of ith sample and c is the number of classes. The values of $Y_{i,j}$ indicates the association between the ith sample and the jth label. If label j is assigned to the ith sample , then $Y_{i,j} = 1$, otherwise $Y_{i,j} = -1$. Particularly, $Y_{i,j} = 0$ denotes the missing label, showing the uncertain association between the ith sample and jth label. To recover the missing labels, we further define a predicted label matrix $\mathbf{F} \in \mathbb{R}^{n \times c}$. We simply set $\mathbf{F}_l = \mathbf{Y}_l$ for the labeled locations, and the value of missing labels are set to zeros.

Missing labels terribly damaged the inherent label structure, making it hard to address true label correlations. Hence, the key issue lies on how to model and capture the relations between labels and features. To recover the missing labels, the general model can formulated as

$$J = \min_{f, \mathbf{F}_l = \mathbf{Y}_l} \sum_{i=1}^{n} loss(f(\mathbf{x}_i), \mathbf{y}_i) + \mu \Omega(f) \tag{1}$$

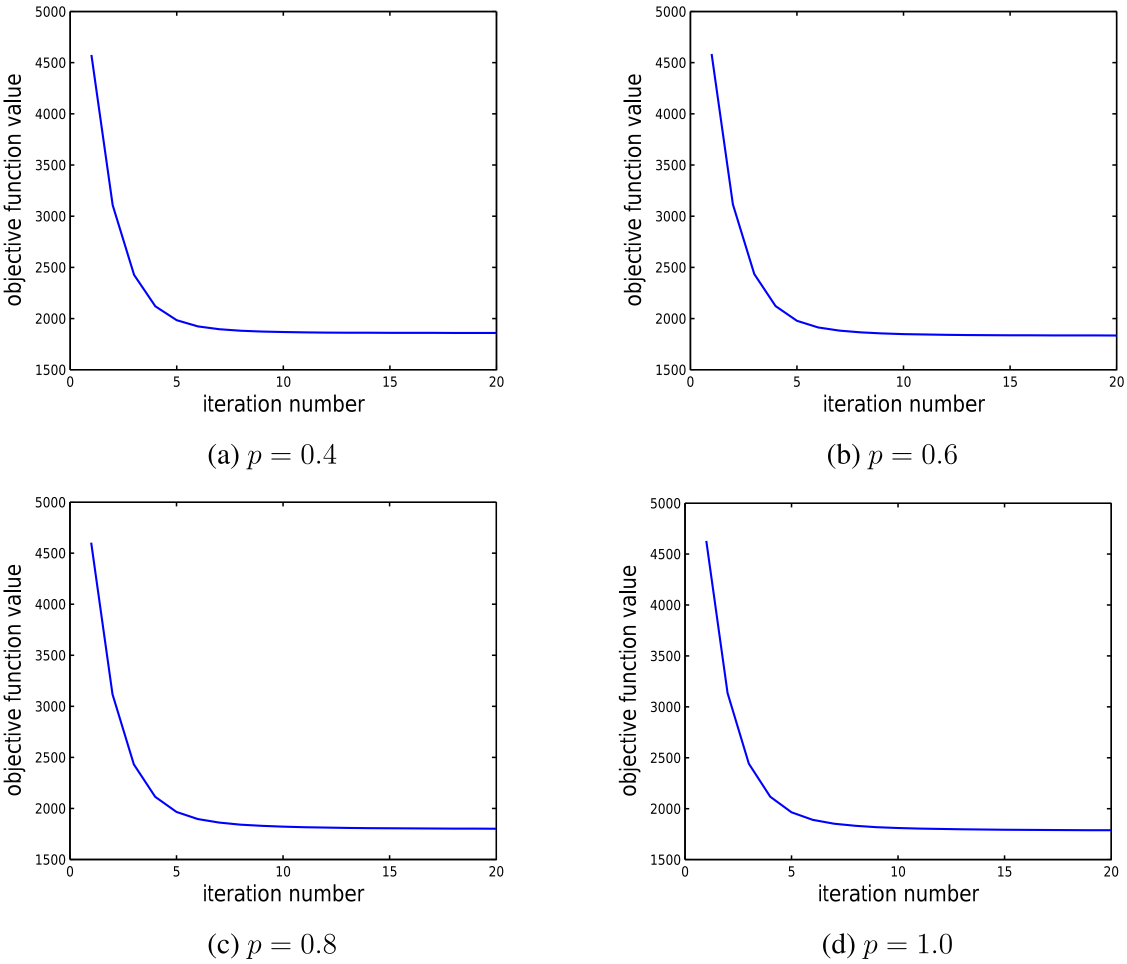
where *loss* $(\cdot)$ is a loss function, $\Omega(f)$ is the regularization term, and μ is a positive constant. The model aims to minimize the loss between the predicted labels and the original labels. Additionally, an regularization term is introduced to avoid overfitting.

Recently, researchers have developed many models to recover missing labels, e.g., matrix completion, label propagation [13], deep learning [34], etc. However, there are few works that focus on the curse of dimensionality in multi-label classification with missing labels.

4. The proposed model

4.1. Motivation

With the incomplete label information, traditional multi-label feature selection algorithms which require complete labels can not address the inherent label structure and correlations. Hence, we expect to propose an effective model to deal with these problems. Here, we utilize the linear regression model to connect feature matrix and label matrix for simplicity and robustness. To preserve the



(a) $p = 0.4$

(b) $p = 0.6$

(c) $p = 0.8$

(d) $p = 1.0$

Fig. 2. The convergence curve of MLMLFS with different p values.

local structure, we introduce a manifold regularization so that similar samples in the feature space should stay close in the recovered label space. The proposed model belongs to the embedding feature selection methods. During the process of model construction, missing labels are recovered and the most informative features are selected. It can be formulated as:

$$\left\{\begin{array}{l}\min\limits_{\mathbf{W},\mathbf{b},\mathbf{Y}} \left\|\mathbf{X}^T\mathbf{W}+\mathbf{1}\mathbf{b}^T-\mathbf{Y}\right\|_{2,1}\\ +\alpha\mathrm{tr}((\mathbf{X}^T\mathbf{W}+\mathbf{1}\mathbf{b}^T)^T\mathbf{L}(\mathbf{X}^T\mathbf{W}+\mathbf{1}\mathbf{b}^T))\\ +\lambda\|\mathbf{W}\|_{2,p}^{p}\end{array}\right\} \tag{2}$$

where $\mathbf{X}$ is the data matrix and $\mathbf{Y}$ is the label matrix with randomly missing labels. $\mathbf{1}\in\mathbb{R}^{n\times 1}$ is a column vector with all elements being 1. Instance level label dependency $\mathrm{tr}((\mathbf{X}^T\mathbf{W}+\mathbf{1}\mathbf{b}^T)^T\mathbf{L}(\mathbf{X}^T\mathbf{W}+\mathbf{1}\mathbf{b}^T)$ requires that similar instances have similar labels. $\mathbf{L}$ denotes the Laplacian matrix defined by $\mathbf{L}=\mathbf{D}-\mathbf{S}$, where $\mathbf{S}$ is a weight matrix and $\mathbf{D}$ is a diagonal matrix. This manifold regularization item preserves the local data structure. $L_{2,\,1}$-norm loss is introduced to measure the effect of outliers and promote robustness. Besides, a $l_{2,\,p}$-norm regularizer is imposed on $\mathbf{W}$ to select the discriminative and representative features. Here, we give the definition of $\|\mathrm{W}\|_{2,p}^{p}$ [35].

$$\|\mathrm{W}\|_{2,p}^{p}=\sum_{i=1}^{d}\left(\sum_{j=1}^{c}w_{ij}^{2}\right)^{p/2}=\sum_{i=1}^{d}\|w_i\|^{p} \tag{3}$$

where w_i is the ith row of $\mathbf{W}$, d and c are the number of features and classes, respectively. The feature importance can be evaluated by $\|w_i\|_2$. For example, when $\|w_i\|_2$ is zero, it means the ith feature can be negligible. Thus, feature selection can be implemented by ranking features with $\|w_i\|_2$. Note that the sparsity on $\mathbf{W}$ increases with the value of p decreasing.

4.2. Optimization and algorithms

There are two variables including $\mathbf{W}$ and $\mathbf{b}$ in our objective function. The optimization problem is convex but non-smooth when $p=1$. When $0<p<1$, the problem is non-convex and it is hard to solve. In this paper we propose an iterative reweighted least square algorithm to solve the problem in Eq. (2). Firstly, we rewrite the objective function into the equivalent formulation.

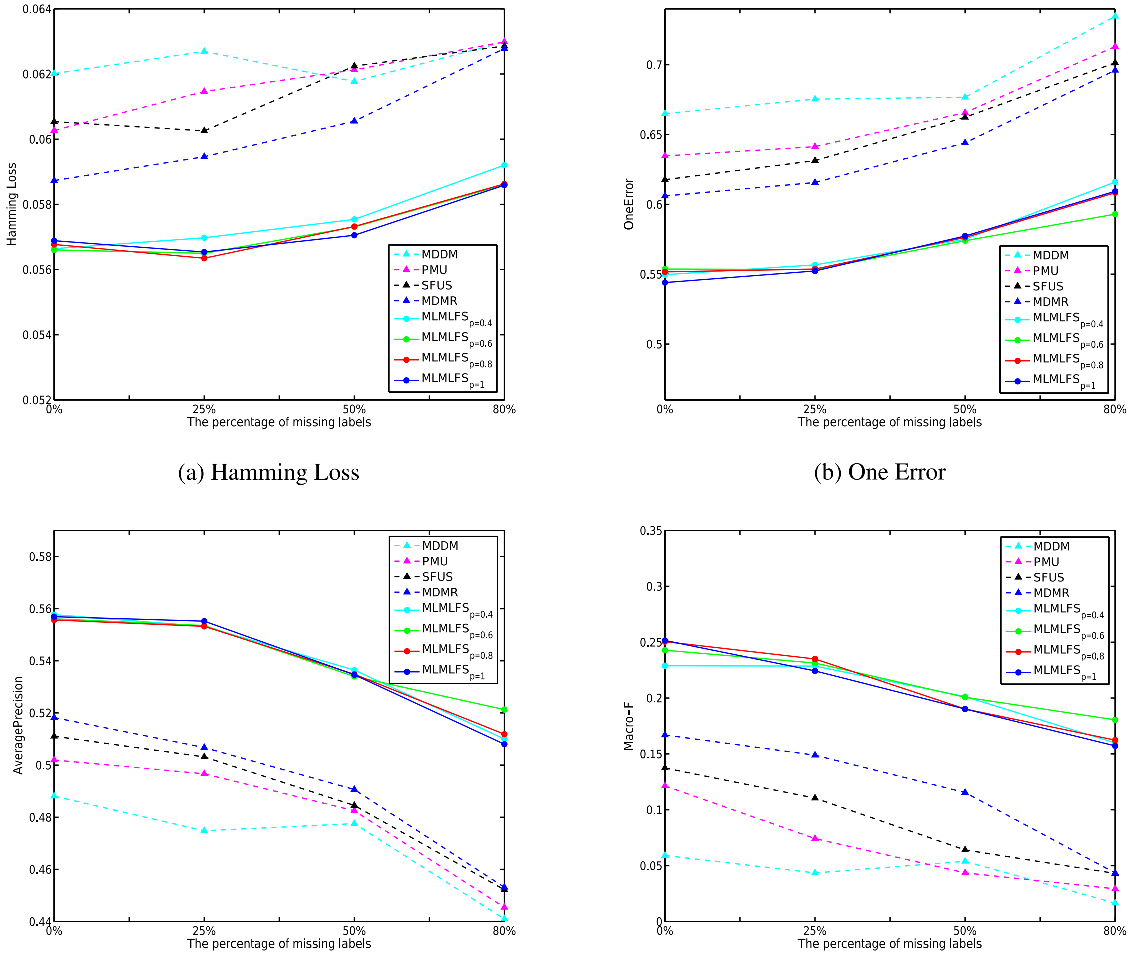
$$\min_{\mathbf{W},\mathbf{b},\mathbf{Y}}\left\{\begin{array}{l}tr((\mathbf{X}^T\mathbf{W}+\mathbf{1}\mathbf{b}^T-\mathbf{Y})^T\mathbf{G}_0(\mathbf{X}^T\mathbf{W}+\mathbf{1}\mathbf{b}^T-\mathbf{Y}))\\ =\alpha\mathrm{tr}((\mathbf{X}^T\mathbf{W}+\mathbf{1}\mathbf{b}^T)^T\mathbf{L}(\mathbf{X}^T\mathbf{W}+\mathbf{1}\mathbf{b}^T))+\lambda\|\mathbf{W}\|_{2,p}^{p}\end{array}\right\} \tag{4}$$

$\mathbf{G}_0=\frac{1}{2\|x_i^T\mathbf{W}+b_i^T-y_i\|}$ is a diagonal matrix with respect to the $l_{2,\,1}$-norm. Here, x_i, b_i and y_i are the ith row of $\mathbf{X}$, $\mathbf{b}$ and $\mathbf{Y}$, respectively.

After that, we can set the derivative of Eq. (4) *w.r.t.* $\mathbf{b}$ to 0 by omitting the irrelevant items, and we have

$$\mathbf{b}=\frac{1}{m}\mathbf{Y}^T\mathbf{G}_0\mathbf{1}-\frac{1}{m}\mathbf{W}^T\mathbf{X}(\mathbf{G}_0+\alpha\mathbf{L}^T)\mathbf{1} \tag{5}$$

where $m=\mathbf{1}^T\mathbf{G}_0\mathbf{1}+\alpha\mathbf{1}^T\mathbf{L}\mathbf{1}$. Now, we have the solution with respect to $\mathbf{b}$.



(a) Hamming Loss

(b) One Error

(c) Average Precision

(d) Macro-F

Fig. 3. Performance comparison on dataset *Artificial*.

To obtain the solution of **W**, we can substitute the solution of **b** into Eq. (4), the optimization problem becomes

$$\left\{\begin{array}{l}\min\limits_{\mathbf{W},\mathbf{b},\mathbf{Y}} tr((\mathbf{NX}^T\mathbf{W}-\mathbf{HY})^T\mathbf{G}_0(\mathbf{NX}^T\mathbf{W}-\mathbf{HY}))\\ +\alpha\, tr((\mathbf{NX}^T\mathbf{W}-(\mathbf{H}-1)\mathbf{Y})^T\mathbf{L}(\mathbf{NX}^T\mathbf{W}-(\mathbf{H}-1)\mathbf{Y}))\\ +\lambda\|\mathbf{W}\|_{2,p}^p\end{array}\right\} \tag{6}$$

where **H** is a centering matrix with $\mathbf{H}=\mathbf{H}^T$. $\mathbf{H}=\mathbf{I}-\frac{1}{m}\mathbf{1}\mathbf{1}^T\mathbf{G}_0$, $\mathbf{N}=\mathbf{I}-\frac{1}{m}\mathbf{1}\mathbf{1}^T(\mathbf{G}_0+\alpha\mathbf{L})$ and **I** is an identity matrix.

As the optimization of W is a linear regression problem involved with $l_{2,\,p}$-norm, we proposed to use iterative reweighted least squares (IRLS) algorithm to solve it. Given the current $\mathbf{W}^t$, the diagonal weighting matrices $\mathbf{G}_1^t$ is defined as:

$$g_j^t=\frac{p}{2}\|w_j^t\|_2^{p-2} \tag{7}$$

Here, g_j^t is the jth diagonal element of $\mathbf{G}_1^t$ and w_j^t is the jth row of $\mathbf{W}^t$. To find the parameters $\mathbf{W}=(\mathbf{W}^1,\cdots,\mathbf{W}^t)^T$ which minimize the $l_{2,\,p}$-norm for the linear regression problem, the IRLS algorithm at step $t+1$ involves solving the following weighted linear least squares problem.

$$\mathbf{W}^{t+1}=\arg\min_{\mathbf{W}} Q(\mathbf{W}|\mathbf{W}^t)=$$
$$\left\{\begin{array}{l}\arg\min\limits_{\mathbf{W}} tr((\mathbf{NX}^T\mathbf{W}-\mathbf{HY})^T\mathbf{G}_0(\mathbf{NX}^T\mathbf{W}-\mathbf{HY}))\\ +\alpha\, tr((\mathbf{NX}^T\mathbf{W}-(\mathbf{H}-1)\mathbf{Y})^T\mathbf{L}(\mathbf{NX}^T\mathbf{W}-(\mathbf{H}-1)\mathbf{Y}))\\ +\lambda tr(\mathbf{W}^T\mathbf{G}_1^t\mathbf{W})\end{array}\right\} \tag{8}$$

By setting $\frac{\partial Q(\mathbf{W}|\mathbf{W}^t)}{\partial \mathbf{W}}=0$, $\mathbf{K}=\mathbf{XN}^T$, we arrive at the solution of $\mathbf{W}^{t+1}$.

$$\mathbf{W}^{t+1}=(\mathbf{K}(\mathbf{G}_0+\alpha\mathbf{L})\mathbf{K}^T+\lambda\mathbf{G}_1^t)^{-1}\mathbf{K}(\mathbf{G}_0\mathbf{H}+\alpha\mathbf{L}(\mathbf{H}-1))\mathbf{Y} \tag{9}$$

As for Eq. (9), the time complexity of computing $(\mathbf{K}(\mathbf{G}_0+\alpha\mathbf{L})\mathbf{K}^T+\lambda\mathbf{G}_1^t)^{-1}$ is $O(d^3)$. As for some feature selection tasks, the feature dimensions are normally very high. That is, we have large d value, leading to extremely high time complexity. Under this circumstance, we propose to reduce the computation burden by introducing the Woodbury Matrix Identity.

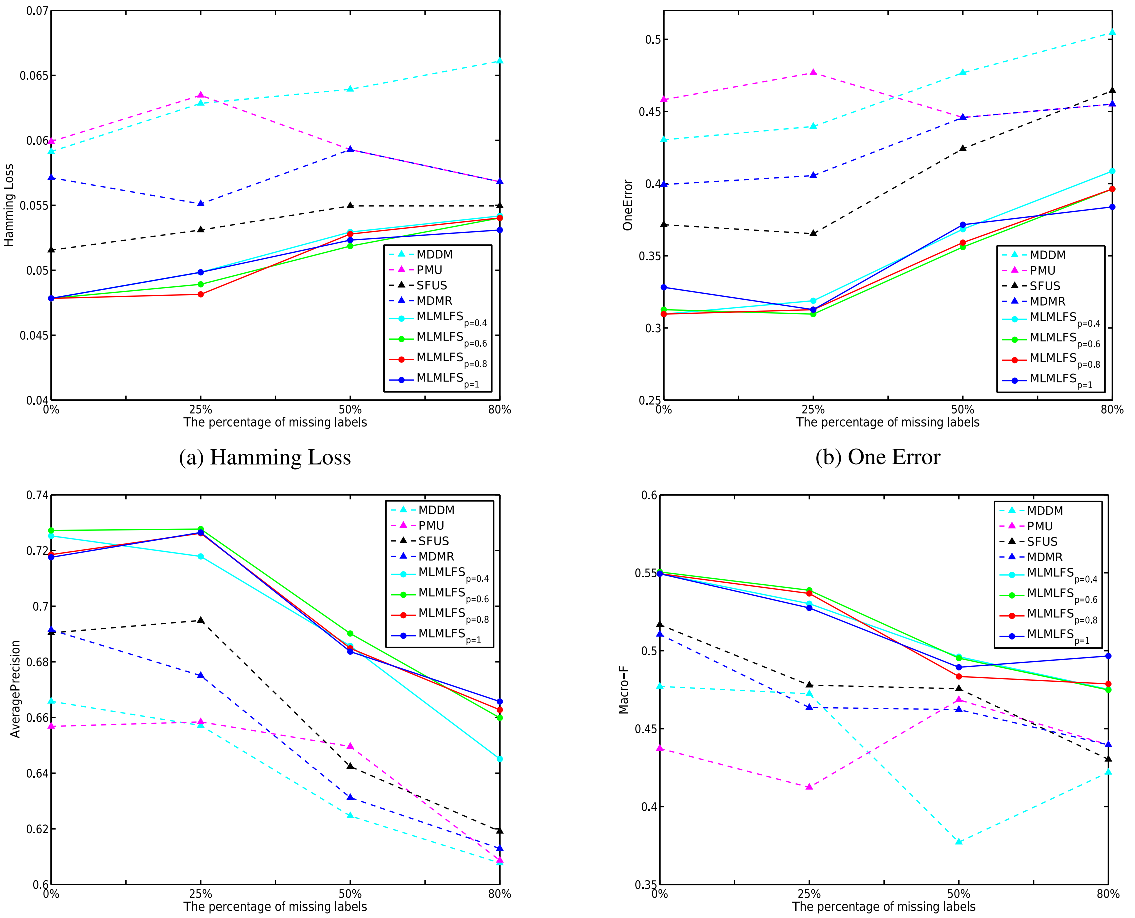
$$(\mathbf{A}+\mathbf{BCD})^{-1}=\mathbf{A}^{-1}-\mathbf{A}^{-1}\mathbf{B}(\mathbf{C}^{-1}+\mathbf{DA}^{-1}\mathbf{B})^{-1}\mathbf{DA}^{-1} \tag{10}$$

With Woodbury Matrix Identity, Eq. (9) can be converted into the following equivalent form.

$$\mathbf{W}^{t+1}=\left\{\begin{array}{l}(\mathbf{G}_1^t)^{-1}\mathbf{K}(\lambda\mathbf{I}+(\mathbf{G}_0+\alpha\mathbf{L})\mathbf{K}^T(\mathbf{G}_1^t)^{-1}\mathbf{K})^{-1}\\ (\mathbf{G}_0\mathbf{H}+\alpha\mathbf{L}(\mathbf{H}-1))\mathbf{Y}\end{array}\right\} \tag{11}$$

Hence, we can update $\mathbf{W}^{t+1}$ according to the relationship between the feature dimension d and the sample number n. If the $d<n$, we can directly compute **W** using Eq. (9). Otherwise, we should use Eq. (11).

After the updating of the feature selection matrix **W** , we update the diagonal weighting matrix $\mathbf{G}_1^t$ by Eq. (7). We introduce a sufficiently small tolerance value to the updating of $\mathbf{G}_1^t$ to get a



(a) Hamming Loss

(b) One Error

(c) Average Precision

(d) Macro-F

Fig. 4. Performance comparison on dataset *Birds*.

stale solution.

$$g_j^t = \frac{p}{2\max\left(\left\|w_j^t\right\|_2^{2-p}, \varepsilon\right)} \tag{12}$$

The optimal stable solution can be got by iteratively updating **b** and **W**, respectively.

In each iteration, the values of missing elements in the prediction label matrix $\tilde{\mathbf{F}} = \mathbf{X}^T\mathbf{W} + \mathbf{1b}^T$ also need to be adjusted.

$$F_{ij} = \begin{cases} -1, & if\ \tilde{F}_{ij} \le -1 \\ \tilde{F}_{ij}, & if\ -1 < \tilde{F}_{ij} < 1 \\ 1, & if\ \tilde{F}_{ij} \ge 1 \end{cases} \tag{13}$$

Here, the value of F_{ij} is set in the range of −1 to 1 to avoid a trivial solution. By iteratively updating **b**, $\mathbf{W}_t$, $\mathbf{G}_0$ and $\mathbf{G}_1^t$, the objective value of Eq. (2) monotonically decreases and guarantees to converge to a fixed point. After that, with the linear projection matrix **W** and the bias vector **b**, we can arrive at the final stable predicted label matrix **F**. To get the discrete label values, we set $F_{i,j} = 1$ if $F_{i,\,j} > 0$ and $F_{i,j} = -1$ if $F_{i,\,j} < 0$ at the end of iteration.

As for the selection of features, we have imposed the sparsity constraint on the feature weight matrix $\mathbf{W} \in \mathbb{R}^{d\times n}$ and the feature importance can be evaluated by $\|w_i\|$, $i \in 1, \cdots, d$. Similar with unsupervised feature selection methods, to select the most significant features, we calculate all $\|w_i\|$ values and sort them in descending order. Then, we can select the first few features according to our demanding.

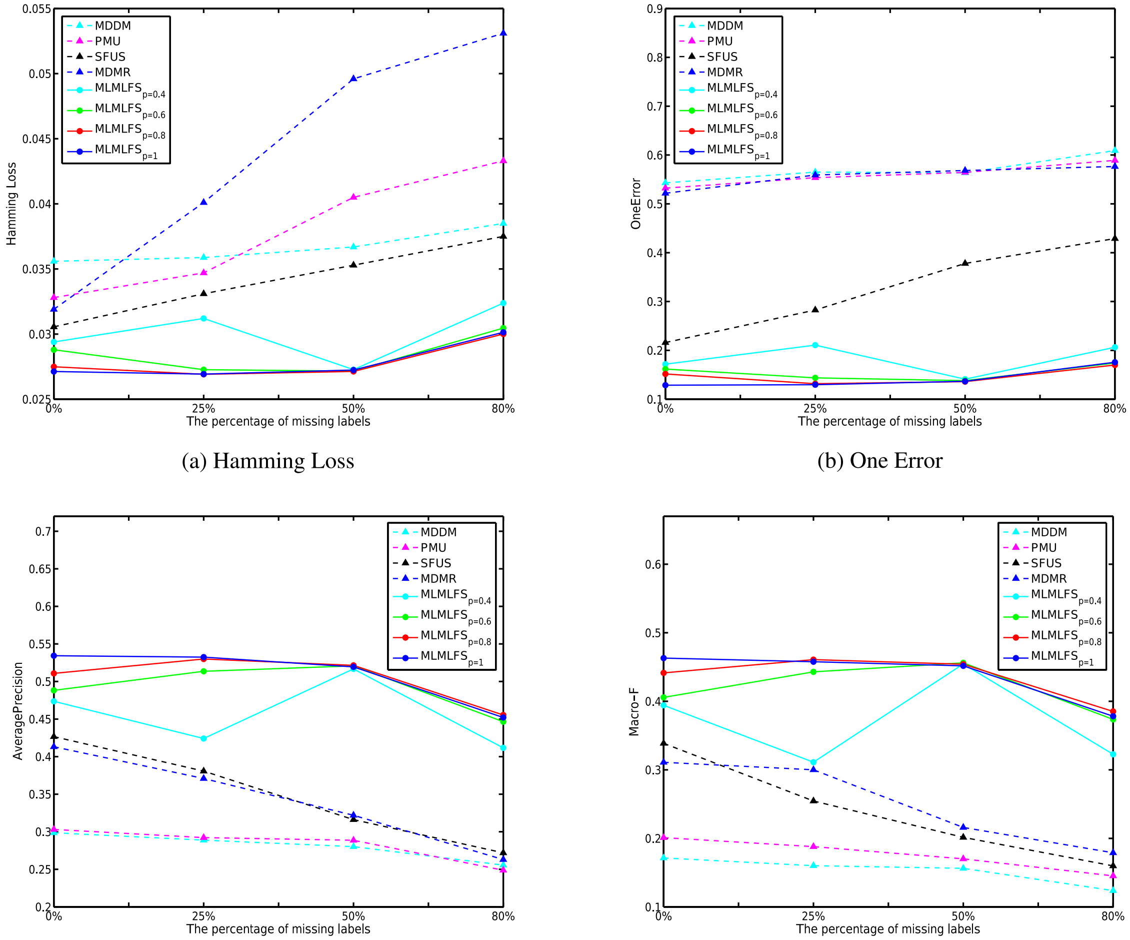
Hence, the missing labels are recovered via the linear regression and the most informative features are selected by the projection matrix. We further summarize the steps of MLMLFS in Algorithm 1.

Algorithm 1 Robust multi-label feature selection with missing labels (MLMLFS).

```
Input:
    Training data $\mathbf{X} \in \mathbb{R}^{n\times d}$
    Training label matrix with missing labels $\mathbf{Y} \in \mathbb{R}^{n\times c}$
    Parameter $\alpha$, $\lambda$, $p$, missing percent $h$
Set $t = 0$ and initialize $\mathbf{W^0} \in \mathbb{R}^{d\times c}$;
repeat
    Compute the diagonal matrices $\mathbf{G}_0$ and $\mathbf{G}_1$;
    if $d < n$ then
        $\mathbf{W}^{t+1}$ can be updated by Eq. (9);
    else
        $\mathbf{W}^{t+1}$ can be updated by Eq. (11).
    end if
    Compute $\mathbf{b}^{t+1}$ according to Eq. (5);
    Compute $\tilde{\mathbf{F}}^{t+1}$ according to $\tilde{\mathbf{F}}^{t+1} = \mathbf{X}^T\mathbf{W}^{t+1} + \mathbf{1b}^T$;
    Adjust $\mathbf{F}$ according to Eq. (13);
until Convergence criterion satisfied.
Output:
    Feature selection matrix $\mathbf{W} \in \mathbb{R}^{d\times c}$
    Predicted label matrix $\mathbf{F} \in \mathbb{R}^{n\times c}$
```

4.3. Time complexity

There are mainly two variables including **b** and **W** in our model. In each iteration, the solution of **b** can be directly computed by Eq. (5). While for the updating of the projection matrix **W**, we



(a) Hamming Loss

(b) One Error

(c) Average Precision

(d) Macro-F

Fig. 5. Performance comparison on dataset *Bookmarks*.

need to iteratively computing $\mathbf{W}^t$, which contributes to the main computations. Generally speaking, in many multi-label learning tasks, the number of labels c are relatively smaller compared with feature dimension d and sample number n. Hence, we mainly focus on the numerical relationship between d and n and give the complexity analysis under different circumstances We denote T as the iteration number. According to the previous discussion, when $d < n$, the updating of $\mathbf{W}^t$ can be got by Eq. (9), and the time complexity is $O(Tnd^2)$. On the contrary, when we have $d > n$, the updating of $\mathbf{W}^t$ can be got by Eq. (11) with the help of Woodbury Matrix Identity. Hence, the time complexity becomes $O(Tn^2d)$.

4.4. Convergence analysis

As for our objective function in Eq. (2), when $p = 1$, we have the standard $l_{2,1}$-norm constraint and it is convex [36]. When $0 < p < 1$, as the problem is non-convex, we proposed to solve the problem by using iterative reweighted least squares (IRLS). Here, we will give the detailed analysis to prove that the proposed model can converge to a stationary point.

For the model in Eq. (2), when $p = 1$, the objective function is convex. The convergence of the $l_{2,1}$-norm optimization problem has been well studied and proved [36]. Hence, when $p = 1$, the model in Eq. (2) is sure to converge. When $0 < p < 1$, we will prove the proposed model can converge to a stationary point.

Suppose after the tth iteration, we obtain $\mathbf{W}^t$, $\mathbf{b}^t$ and $\mathbf{F}^t$. With respect to the updating of $\mathbf{W}^{t+1}$, we fix the value of $\mathbf{F}$ as $\mathbf{F}^t$ and $\mathbf{b}$ as $\mathbf{b}^t$. Let

$$L(\mathbf{W}) = \left\{ \begin{array}{l} \left\|\mathbf{X}^T\mathbf{W} + \mathbf{1}\mathbf{b}^T - \mathbf{Y}\right\|_{2,1} \\ +\alpha \mathrm{tr}((\mathbf{X}^T\mathbf{W} + \mathbf{1}\mathbf{b}^T)^T\mathbf{L}(\mathbf{X}^T\mathbf{W} + \mathbf{1}\mathbf{b}^T)) \\ +\lambda\|\mathbf{W}\|_{2,p}^p \end{array} \right\} \tag{14}$$

In the following, we will prove that $L(\mathbf{W})$ can be minimized by iteratively minimizing $Q(\mathbf{W}|\mathbf{W}^t.)$.

Theorem 1. *$L(\mathbf{W}) - Q(\mathbf{W}\big|\mathbf{W}^t)$ attains its maximum when $\mathbf{W} = \mathbf{W}^t$.*

Proof. Let $F(\mathbf{W}) = L(\mathbf{W}) - Q(\mathbf{W}\big|\mathbf{W}^t)$. Firstly, we will prove that as for $\forall\mathbf{W}$, there is $F(\mathbf{W}^t) - F(\mathbf{W}) \geq 0$. $F(\mathbf{W}^t)$ can be written as

$$\begin{aligned} F(\mathbf{W}^t) &= L(\mathbf{W}^t) - Q(\mathbf{W}^t\big|\mathbf{W}^t) \\ &= \lambda(1 - \tfrac{p}{2})\textstyle\sum_{i=1}^d \left\|w_j^t\right\|_2^p \end{aligned} \tag{15}$$

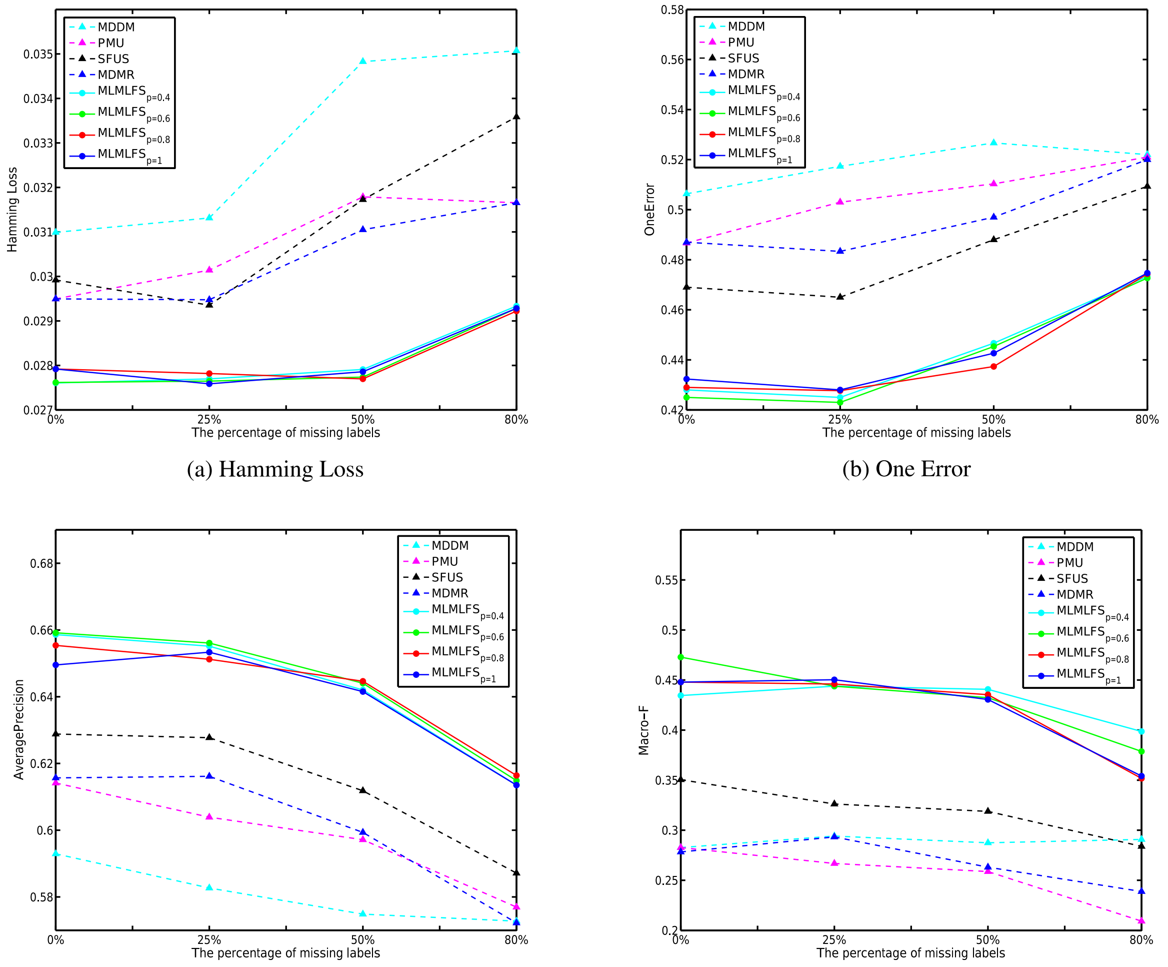
$F(\mathbf{W})$ can be written as

$$\begin{aligned} F(\mathbf{W}) &= L(\mathbf{W}) - Q(\mathbf{W}\big|\mathbf{W}^t) \\ &= \lambda\textstyle\sum_{i=1}^d \left(\left\|w_j\right\|_2^p - \tfrac{p}{2}\frac{\|w_j\|_2^2}{\|w_j^t\|_2^{2-p}}\right) \end{aligned} \tag{16}$$

Then we can have

$$F(\mathbf{W}^t) - F(\mathbf{W}) =$$

$$\lambda\textstyle\sum_{i=1}^d \left((1 - \tfrac{p}{2})\left\|w_j^t\right\|_2^p - \left\|w_j\right\|_2^p + \tfrac{p}{2}\frac{\|w_j\|_2^2}{\|w_j^t\|_2^{2-p}}\right) \tag{17}$$



(a) Hamming Loss

(b) One Error

(c) Average Precision

(d) Macro-F

Fig. 6. Performance comparison on dataset *Reference*.

Let $a = \left\|w_j^t\right\|_2$ and $b = \left\|w_j\right\|_2$, we can define $H_j = (1-\frac{p}{2})\left\|w_j^t\right\|_2^p - \left\|w_j\right\|_2^p + \frac{p}{2}\frac{\left\|w_j\right\|_2^2}{\left\|w_j^t\right\|_2^{2-p}}$ and it can be written as

$$H_j(b) = (1-\frac{p}{2})a^p - b^p + \frac{p}{2}a^{p-2}b^2 \tag{18}$$

where $H_j(b)$ is a polynomial function about b. We take the first and second order derivatives of H_j w.r.t. b

$$H_j{}'(b) = p(a^{p-2}b - b^{p-1}) \tag{19}$$

$$H_j{}''(b) = p(a^{p-2} - (p-1)b^{p-2}) \tag{20}$$

With $b \geq 0$, $a \geq 0$ and $0 < p < 1$, when $b = a$, we can get $H_j(a) = 0$, $H_j{}'(a) = 0$ and $H_j{}''(a) = p(2-p)a^{p-2} > 0$. Hence, we have $H_j(b) \geq 0$ always holds. Therefore, $F(\mathbf{W}^t) - F(\mathbf{W}) = \lambda\sum_{i=1}^{d} H_j \geq 0$ and $F(\mathbf{W}) = L(\mathbf{W}) - Q(\mathbf{W}|\mathbf{W}^t)$ attains its maximum when $\mathbf{W} = \mathbf{W}^t$.

Theorem 2. *Let* $\mathbf{W}^{t+1} = \arg\min_{\mathbf{W}} Q(\mathbf{W}|\mathbf{W}^t)$. $L(\mathbf{W}^{t+1}) \leq L(\mathbf{W}^t)$ *always holds.*

Proof. According to the above theorem, we can easily have

$$\begin{aligned} L(\mathbf{W}^{t+1}) &= L(\mathbf{W}^{t+1}) - Q(\mathbf{W}^{t+1}|\mathbf{W}^t) + Q(\mathbf{W}^{t+1}|\mathbf{W}^t) \\ &\leq L(\mathbf{W}^t) - Q(\mathbf{W}^t|\mathbf{W}^t) + Q(\mathbf{W}^{t+1}|\mathbf{W}^t) \\ &\leq L(\mathbf{W}^t) - Q(\mathbf{W}^t|\mathbf{W}^t) + Q(\mathbf{W}^t|\mathbf{W}^t) \\ &= L(\mathbf{W}^t) \end{aligned} \tag{21}$$

Therefore, in each iteration, $\mathbf{W}^{t+1}$ can be updated by minimizing $Q(\mathbf{W}^{t+1}|\mathbf{W}^t)$ and $\mathbf{b}^{t+1}$ can be updated by Eq. (5). $\mathbf{F}^{t+1}$ can be updated by $\tilde{\mathbf{F}}^{t+1} = \mathbf{X}^T\mathbf{W} + \mathbf{1}\mathbf{b}^T$ and Eq. (13). Here, we have proved the proposed method can surely converge to a stationary point.

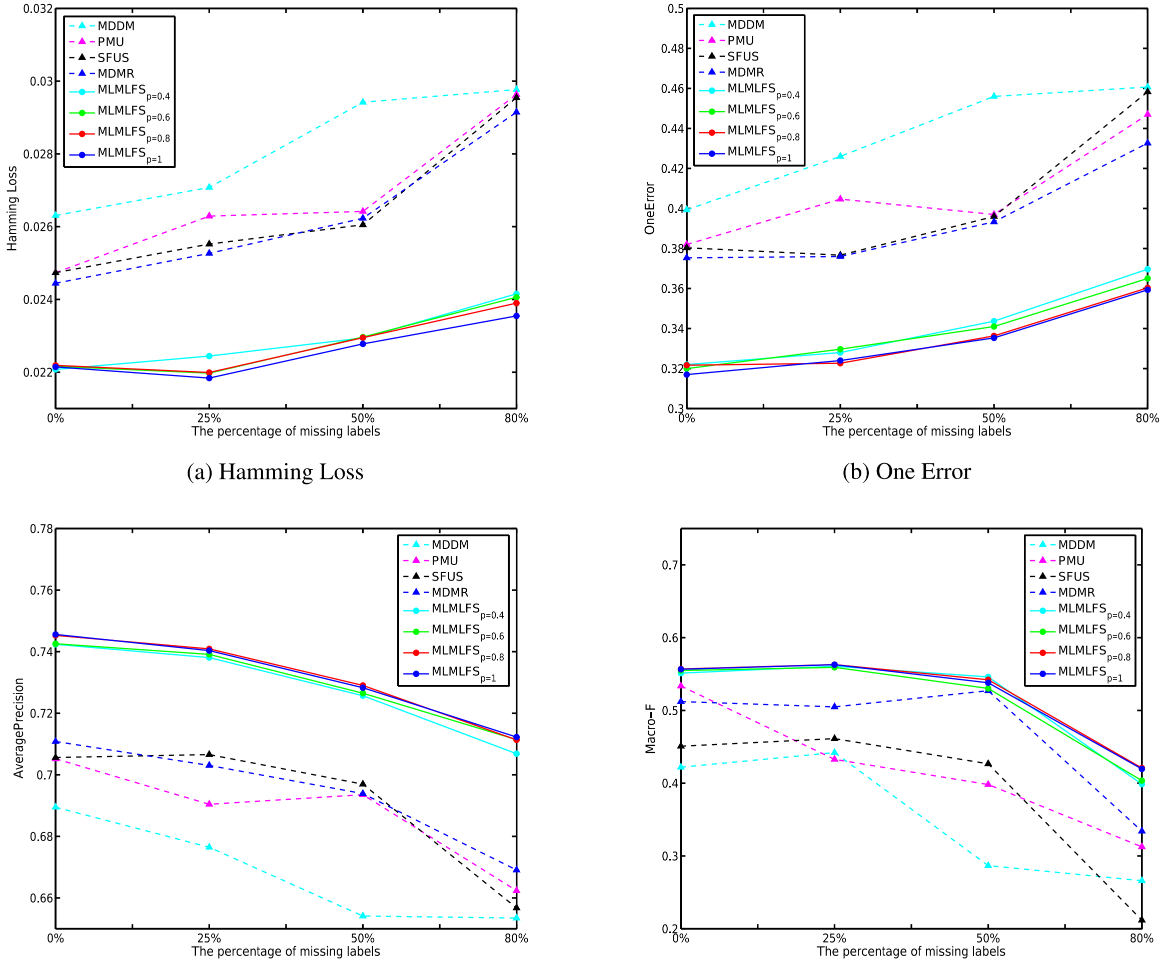
We fix the value of λ and α, and get the convergence curves of MLMLFS with different p values in Fig. 2. Dataset *Reference* with 50% labels randomly missing is used here. We can see that MLMLFS converges rapidly with different p values.

5. Experimental analysis

To verify the effectiveness of our proposed method, we conduct experiments on six multi-label datasets under various missing percentages. The performances of our method and the comparison algorithms are illustrated by tables and figures. At the end of this section, we give an discussion about the results.

5.1. Datasets and evaluation metrics

To validate the effectiveness of our model, we use six benchmark datasets, i.e., Artificial, Birds, Bookmarks, Reference, Social and Yeast. The details about the six datasets are presented in Table 1.



(a) Hamming Loss

(b) One Error

(c) Average Precision

(d) Macro-F

Fig. 7. Performance comparison on dataset *Social*.

Among the six selected datasets, *Artificial, Reference* and *Social* are from Yahoo which collects multi-label text categorization data from “yahoo.com” [37]. The remaining datasets *Birds, Bookmarks* and *Yeast* are from Mulan [38], an open-source Java library for multi-label learning. Considering that our task mainly involves with feature selection, the number of selected features varies from 103 to 2150.

Compared with single-label learning tasks, evaluation criteria used for multi-label ones are more complicated for one instance can be assigned to several labels simultaneously. Here, we use four evaluation metrics include *Hamming Loss, One Error, Average Precision* and *Macro-F* to evaluate the performance of multi-label feature selection. We denote T as the test set and $T = \{(x_i, Y_i)|1 \le i \le N\}$. Let h: $X \to Y$ be a classification hypothesis. For instance $x_i \in X$, $h(x_i)$ gives the predicted label vector. We give the definition of the above evaluation metrics as below.

(1) *Hamming Loss* evaluates how many times an instance-label pair is misclassified.

$$HammingLoss_T(h) = \frac{1}{N}\sum_{i=1}^{N}\frac{|h(x_i) \oplus y_i|}{C} \tag{22}$$

where $h(x_i) \oplus y_i$ denotes the XOR operation between the predicted label vector $h(x_i)$ and the ground-truth label vector y_i.

(2) *One Error* evaluates how many times the top-ranked label is not in the set of ground-truth labels of the instances.

$$OneError_T(f) = \frac{1}{N}\sum_{i=1}^{N}[\![[\underset{y \in Y}{argmax}\ f(x_i, y)] \notin Y_i]\!] \tag{23}$$

where for any predicate x, $[\![x]\!]$ equals 1 if x holds and 0 otherwise.

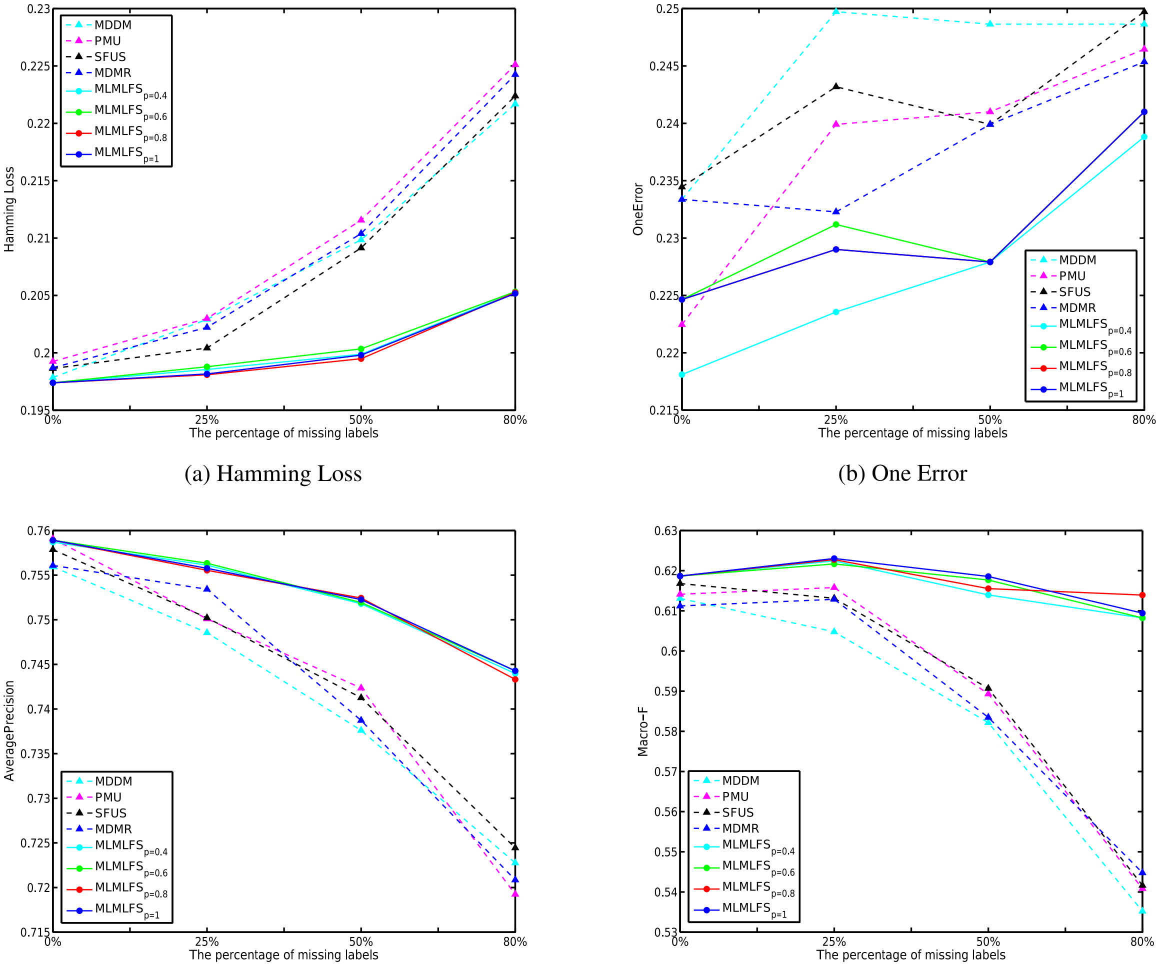
(3) *Average Precision* evaluates the average fraction of labels ranked above a particular label $y \in Y$ which actually are in Y.

$$AveragePrecision_T(f) = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{|Y_i|}\sum_{y \in Y_i}\frac{|\{y' \in Y_i | rank_f(x_i, y') \le rank_f(x_i, y)\}|}{rank_f(x_i, y)} \tag{24}$$

(4) *Macro-F* computes the average fraction of labels ranked above a particular label.

$$MacroF_T(h) = \frac{1}{C}\sum_{l=1}^{C}\frac{2\sum_{i=1}^{N} h^l(x_i) y_i^l}{\sum_{i=1}^{N} y_i^l + \sum_{i=1}^{N} h^l(x_i)} \tag{25}$$

Among the seven evaluation metrics, *Hamming Loss* and *Macro-F* are based on the multi-label classifier $h(\cdot)$, which evaluates how many times an instance-label pair is misclassified. *One Error* and *Average Precision* are based on the real-valued function $f(\cdot,\cdot)$ which evaluates the ranking quality of different labels for each instance. As for *Hamming Loss* and *One Error*, smaller value brings to better performance. While for *Average Precision* and *Macro-F*, bigger value brings to better performance.



(a) Hamming Loss

(b) One Error

(c) Average Precision

(d) Macro-F

Fig. 8. Performance comparison on dataset *Yeast*.

5.2. Comparison methods

Multi-label feature selection has been studied for many years and there are several benchmark algorithms. However, most of the methods assume that the label matrix is complete, while the assumption is usually violated in real-world applications. Since no researchers have explored how to select features when the labels are randomly missing. We proposed a novel method to recover the missing labels and select the most informative features simultaneously. Here, we validate the effectiveness of MLMLFS by comparing with the traditional feature selection algorithms. The detailed information of the four comparison algorithms named MDDM, PMU, SFUS and MDMR is listed below.

- MDDM [10]: It obtains a lower dimensional feature space by maximizing the dependence between the original feature description and the class labels.
- PMU [39]: It considers the mutual information between selected features and the label set.
- SFUS [11]: It is a sparsity-based model and conducts feature selection by capturing the shared subspace of original space for multi-label learning.
- MDMR [5]: It selects features by maximizing the dependency and minimizing the redundancy simultaneously.

5.3. Parameter setting

We have two parameters, i.e., λ for the manifold regularization item and α for the group sparsity constraint. We use grid search strategy to study the influence of parameters on model. The parameter λ is tuned in the range of $\{10^{-6}, 10^{-4}, 10^{-2}, 10^{0}, 10^{2}, 10^{4}, 10^{6}\}$, α in the range of $\{10^{-6}, 10^{-4}, 10^{-2}\}$. For all the methods, the best performance are recorded. On the other hand, the percentage of missing labels has a marked impact on the feature selection performance. Here, the percentage of missing labels is set as 0%, 25%, 50% and 80%. Particularly, when there is 0% missing percentage, i.e. the label matrix is complete, it can be seen as the traditional multi-label feature selection. When the missing percentage increases, the label structure is destroyed to greater extent.

5.4. Feature selection results

Traditional multi-label classifier ML-KNN given by Zhang et al. [40] is used as the evaluation classifier. We make minor modifications on it so that it can deal with the missing labels by simply ignoring them. Since the number of features to be selected is still an open question for the task of feature selection, in our experiments, we give the best scores under certain features for all methods. We present the results measured by Hamming Loss, One Error, Average Precision and Macro-F. The experimental results under 0%, 25%, 50%, 80% missing labels are shown in Tables 2–5, respectively. The performance comparison figures in terms of missing label ratio are shown in Figs. 3–8. With these experimental results, we further give some observations and analysis.

- Under 0% missing labels, it can be seen as the traditional multi-label feature selection with complete labels. We can see that our method gets the best performance in most cases. When there are no labels are missing, MLMLFS utilizes a robust linear regression to model the relation between features and labels. And the effective $l_{2,\,p}$-norm $(0<p\leq 1)$ constraint is imposed on the weight matrix W, which retains the most informative features while discards the redundant ones. Overall, the linear regression and group sparsity contributes much to the remarkable performance of MLMLFS. Hence, MLMLFS which aims to deal with multi-label feature selection can also deal with complete cases, and achieves remarkable results compared with baseline algorithms.
- Under 25%, 50% and 80% missing labels, the inherent label structure and correlations are damaged to some extent, and MLMLFS outperforms other algorithms in most cases. Most traditional multi-label feature selection algorithms need to model the relation between feature space and label space. For example, MDDM maximums the dependence between the original feature description and the class labels, PMU considers the mutual information between selected features and the label set. However, under the missing label circumstance, they are out of capability to model correctly for they take no operations for the damaged label structure. As for MLMLFS, the robust linear regression model aims to learn a stable projection from feature space to label space, and then recovery the missed labels with the learned projection. As a result, the incomplete labels are recovered with feature awareness. Moreover, with the group sparsity, MLMLFS can conduct feature selection during the process of model construction. Experimental results demonstrate that label recovery is of much significance to multi-label feature selection with missing labels.
- We also give the figures demonstrating feature selection performance in terms of the percentage of missing labels. Obviously, more missing label leads to greater damaged label structure. With the missing percentage grows, the performance of traditional methods decreases rapidly. On the whole, the performance of MLMLFS is much better than the comparison methods especially when the percentage of the missing labels is very high.
- In this paper, we use the universal form of group sparsity norm, i.e. $l_{2,\,p}$-norm $(0<p\leq 1)$ to exploit the influence of different p values on feature selection performance. Generally, $l_{2,\,1}$-norm is used in feature selection tasks for its simplicity and efficiency. Theoretically, a smaller p value results in greater sparsity. Hence, different p values with different degree of sparsity would bring to diverse feature selection results. According to the experimental results of MLMLFS, we can observe that the best result does not always lie in $p=1$, and it is hard to tell which p value is the best one. Under most circumstances, $p=0.6,\ p=0.8$ and $p=1$ achieve comparable results and there is not an optimal p value for the task of feature selection. Empirically speaking, 0.6, 0.8 and 1 are good choices to for the setting of p value. We think this finding can give much guidance to the following feature selection tasks.

Overall, MLMLFS consistently outperforms other methods on all the datasets under different percentage of missing labels. And the experimental results also correspond to our theoretical analysis.

6. Conclusions and future work

In this paper, we proposed a feature selection model for multi-label classification with missing labels (MLMLFS). The missing labels are recovered by the robust linear regression, and a discriminant feature subset is selected by the effective $l_{2,\,p}$-norm $(0<p\leq 1)$ constraint simultaneously. The predicted label matrix and parameters of the regression model are iteratively updated until the proposed model converges. MLMLFS performs well due to its robustness, predictability and recoverability. Experimental analysis on several multi-label datasets also validates that MLMLFS is superior to the existing multi-label feature selection algorithms under different missing proportions. In the future, we will consider to model the more complex relation between feature space and label space under missing label circumstance.

References

[1] R.E. Schapire, Y. Singer, Boostexter: a boosting-based system for text categorization, Mach. Learn. 39 (2–3) (2000) 135–168.
[2] M.L. Zhang, Z.H. Zhou, Multilabel neural networks with applications to functional genomics and text categorization, IEEE Trans. Knowl. Data Eng. 18 (10) (2006) 1338–1351.
[3] M.R. Boutell, J. Luo, X. Shen, C.M. Brown, Learning multi-label scene classification, Pattern Recognit. 37 (9) (2004) 1757–1771.
[4] M.L. Zhang, Z.H. Zhou, A review on multi-label learning algorithms, IEEE Trans. Knowl. Data Eng. 26 (8) (2014) 1819–1837.
[5] Y. Lin, Q. Hu, J. Liu, J. Duan, Multi-label feature selection based on max-dependency and min-redundancy, Neurocomputing 168 (2015) 92–103.
[6] I. Guyon, A. Elisseeff, An introduction to variable and feature selection, J. Mach. Learn. Res. 3 (6) (2002) 1157–1182.
[7] A. Jain, D. Zongker, Feature selection: evaluation, application, and small sample performance, IEEE Trans. Pattern Anal. Mach. Intell. 19 (2) (1997) 153–158.
[8] W. Chen, J. Yan, B. Zhang, Z. Chen, Q. Yang, Document transformation for multi-label feature selection in text categorization, in: IEEE International Conference on Data Mining, 2007, pp. 451–456.
[9] K. Yu, S. Yu, V. Tresp, Multi-label informed latent semantic indexing, in: International ACM SIGIR Conference on Research and Development in Information Retrieval, 2005, pp. 258–265.
[10] Y. Zhang, Z.H. Zhou, Multi-label dimensionality reduction via dependence maximization, ACM Trans. Knowl. Discov. Data 4 (3) (2010) 1503–1505.
[11] H. Wang, C. Ding, H. Huang, Multi-label linear discriminant analysis, in: European Conference on Computer Vision, 2010, pp. 126–139.
[12] Z. Ma, F. Nie, Y. Yang, J.R.R. Uijlings, N. Sebe, Web image annotation via subspace-sparsity collaborated feature selection, IEEE Trans. Multimedia 14 (4) (2012) 1021–1030.
[13] B. Wu, S. Lyu, B.G. Hu, Q. Ji, Multi-label learning with missing labels for image annotation and facial action unit recognition, Pattern Recognit. 48 (7) (2015) 2279–2289.
[14] Q. Xu, P. Zhu, Q. Hu, C. Zhang, Robust multi-label feature selection with missing labels, in: Chinese Conference on Pattern Recognition, 2016, pp. 752–765.
[15] T. Joachims, Text categorization with support vector machines: learning with many relevant features, in: Proceedings of European Conferenece on Machine Learning, vol. 1398, 1998, pp. 137–142.
[16] J. Read, B. Pfahringer, G. Holmes, E. Frank, Classifier chains for multi-label classification, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, 2009, pp. 254–269.
[17] A. Elisseeff, J. Weston, A kernel method for multi-labelled classification, International Conference on Neural Information Processing Systems: Natural and Synthetic, 2001, pp. 681–687.
[18] S.S. Bucak, P. Kumar Mallapragada, R. Jin, A.K. Jain, Efficient multi-label ranking for multi-class learning: application to object recognition, IEEE International Conference on Computer Vision, 2010, pp. 2098–2105.
[19] Y. Lin, Q. Hu, J. Zhang, X. Wu, Multi-label feature selection with streaming labels, Inf. Sci. 372 (2016) 256–275.
[20] X. Ding, B. Li, W. Xiong, W. Guo, W. Hu, B. Wang, Multi-instance multi-label learning combining hierarchical context and its application to image annotation, IEEE Trans. Multimedia 18 (8) (2016) 1616–1627.
[21] E.A. Cherman, M.C. Monard, H.D. Lee, Filter approach feature selection methods to support multi-label learning based on relieff and information gain, in: Brazilian Conference on Advances in Artificial Intelligence, 2012, pp. 72–81.
[22] A. Alalga, K. Benabdeslem, N. Taleb, Soft-constrained Laplacian score for semi-supervised multi-label feature selection, Knowl. Inf. Syst. 47 (1) (2015) 1–24.
[23] Y. Lin, Q. Hu, J. Liu, J. Chen, J. Duan, Multi-label feature selection based on neighborhood mutual information, Appl. Soft. Comput. 38 (C) (2016) 244–256.
[24] M.L. Zhang, J.M. Pena, V. Robles, Feature selection for multi-label Naive Bayes classification, Inf. Sci. 179 (19) (2009) 3218–3229.
[25] J. Lee, D.W. Kim, J. Lee, D.W. Kim, Memetic feature selection algorithm for multi-label classification, Inf. Sci. 293 (293) (2015) 80–96.

[26] Q. Gu, Z. Li, J. Han, Correlated multi-label feature selection, in: ACM International Conference on Information and Knowledge Management, 2011, pp. 1087–1096.

[27] X. Cai, F. Nie, W. Cai, H. Huang, New graph structured sparsity model for multi-label image annotations, in: IEEE International Conference on Computer Vision, 2013, pp. 801–808.

[28] Y.-Y. Sun, Y. Zhang, Z.-H. Zhou, Multi-label learning with weak label, in: Twenty-Fourth AAAI Conference on Artificial Intelligence, 2010.

[29] S.S. Bucak, R. Jin, A.K. Jain, Multi-label learning with incomplete class assignments, in: 2013 IEEE Conference on Computer Vision and Pattern Recognition, 2011, pp. 2801–2808.

[30] H. Yu, P. Jain, P. Kar, I. Dhillon, Large-scale multi-label learning with missing labels, in: International Conference on Machine Learning, 2014, pp. 593–601.

[31] M. Chen, A. Zheng, K. Weinberger, Fast image tagging, in: International Conference on Machine Learning, 2013, pp. 1274–1282.

[32] L. Xu, Z. Wang, Z. Shen, Y. Wang, E. Chen, Learning low-rank label correlations for multi-label classification with missing labels, in: 2014 IEEE International Conference on Data Mining, 2014, pp. 1067–1072.

[33] B. Wu, S. Lyu, B. Ghanem, Constrained submodular minimization for missing labels and class imbalance in multi-label learning, in: The Thirtieth AAAI Conference on Artificial Intelligence, 2016.

[34] X. Li, F. Zhao, Y. Guo, Conditional restricted Boltzmann machines for multi-label learning with incomplete labels, in: Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics, 2015, pp. 635–643.

[35] S. Gao, L. Chia, I.W. Tsang, Z. Ren, Concurrent single-label image classification and annotation via efficient multi-layer group sparse coding, IEEE Trans. Multimedia 16 (3) (2014) 762–771.

[36] F. Nie, H. Huang, X. Cai, C. Ding, Efficient and robust feature selection via joint l2,1 -norms minimization, in: International Conference on Neural Information Processing Systems, 2010, pp. 1813–1821.

[37] N. Ueda, Parametric mixture models for multi-labeled text., Adv. Neural Inf. Process. Syst. (2002) 721–728.

[38] G. Tsoumakas, E. Spyromitros-Xioufis, J. Vilcek, I. Vlahavas, Mulan: a java library for multi-label learning, J. Mach. Learn. Res. 12 (7) (2011) 2411–2414.

[39] J. Lee, D.W. Kim, Feature selection for multi-label classification using multivariate mutual information, Pattern Recognit. Lett. 34 (3) (2013) 349–357.

[40] M.L. Zhang, Z.H. Zhou, Ml-knn : a lazy learning approach to multi-label learning, Pattern Recognit. 40 (7) (2007) 2038–2048.

Pengfei Zhu (M'15) received the Ph.D. degree from The Hong Kong Polytechnic University, Hong Kong, China, in 2015. He received his B.S. and M.S. from Harbin Institute of Technology, Harbin, China in 2009 and 2011, respectively. Now he is an associate professor with School of Computer Science and Technology, Tianjin University. His research interests are focused on machine learning and computer vision.

Qian Xu received B.S. degree from Tianjin University, Tianjin, China in 2015. Now she is a master student with School of Computer Science and Technology, Tianjin University. Her research interests are focused on machine learning and data mining.

Qinghua Hu (M'11) received B.S., M.S. and Ph.D. degrees from Harbin Institute of Technology, Harbin, China in 1999, 2002 and 2008, respectively. He was an associate professor with Harbin Institute of Technology from 2008 to 2011. Now he is a full professor with School of Computer Science and Technology, Tianjin University. His research interests are focused on intelligent modeling, data mining, knowledge discovery for classification and regression. He is a PC co-chair of RSCTC 2010 and severs as referee for a great number of journals and conferences. He has published more than 70 journal and conference papers in the areas of pattern recognition and fault diagnosis.

Changqing Zhang (M'15) received the B.S. and M.E. degrees in computer science from Sichuan University in 2005 and 2008, and the Ph.D. degree from Tianjin University in 2016, respectively. He is currently an Assistant Professor with the School of Computer Science and Technology, Tianjin University. His current research is multi-view learning.

Hong Zhao is now a Postdoc in Tianjin University. Her research interest is machine learning.