# Partial Multi-Label Learning via Multi-Subspace Representation

Guangfei Liang

### Abstract

The existing Partial Multi label learning methods seldomly focus on the noise in feature space. To adress this problem, this paper proposes a novel framework named partial multilabel learning via MUlti-SubspacE Representation (MUSER), which take the advantage of the label discriminant information and avoids the noise from feature space. However, the optimization in this paper does NOT statisfy the orthogonal constraint in fact. In this report, we optimize the objective function in another way, which change the update order of variables and use a LDA-like problem to solve the orthogonal constraint. Furthermore, experiments in this paper are repeated to show the proposed model's superior performance.

**Keywords:** Partial multilabel learning, Latent subspace, Orthogonal constraint

## 1 Introduction

The task of Partial Multi-Label Learning (PML) is to learn the precise labels from the samples with redundant labels (Figure.1). Using off-the-shelf MLL methods to train the model [1] is a simple solution. However, the presence of redundant noise labels in training data will degrade performance.

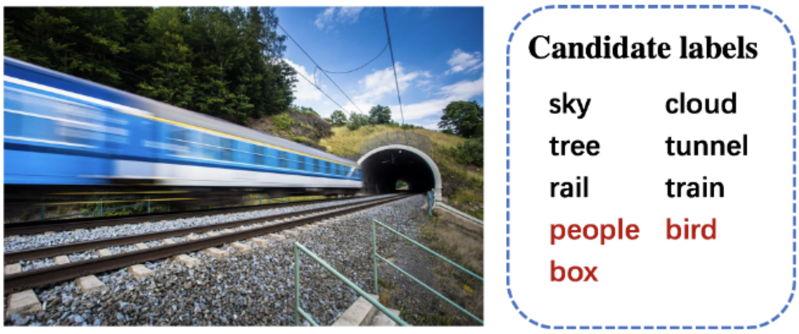


Figure 1: An example of partial multi-label learning with noisy features. Among nine candidate labels of the example, six in black font are ground-truth labels while three in red font are noisy labels. Obviously, the noisy features derived from the high speed motion.

To address the problem, research [2] proposed the first PML framework, which provided a practical solution for dealing with redundant candidate labels. Existing PML methods can be divided into two groups: unified strategy and two-stage strategy. For methods based on unified strategies, the prediction model is learned while simultaneously optimizing candidate labels. By minimizing the ranking loss and taking advantage of data structure, PML-$fp$ and PML-$lc$ [2] optimized label confidence values and trained the model. For training prediction models, fPML [3] adopted a feature and label coherent matrix to factorize the original matrix. Low-rank and sparse decomposition was used by PML-LRS [4] to obtain the ground-truth labels and train the model at the same time. For two-stage methods, the whole training process is divided into two stages, including reliable label selection by disambiguating strategy and model training by using the reliable labels. PARTICLE [5] used iterative label propagation to extract credible labels with high confidence values, which were then used to train the prediction model. To obtain the trustworthy labels with high confidence values, DRAMA [6] performed the feature manifold and introduced a gradient boost model for training.

Obviously, existing PML methods mainly focus on the noise in label space while the noise concealed in feature space is regrettably ignored. To adress this problem, this paper proposes a novel robust PML model named **partial multi-label learning via MUltiSubspacE Representation(MUSER)**, which fuses the feature mapping and label decomposition to train the desired model. Firstly, the original label space is decomposed into a latent label subspace and a label correlation matrix. Secondly, a graph Laplacian regularization is introduced to keep the manifold structure of data. Thirdly, a orthogonal feature selection matrix is used to resist the noise from original feature space. Finally, the unified model is optimized by applying a alternative iteration algorithm.

However, the optimization of this paper can NOT ensure the orthogonal constraint of the feature selection matrix. We slightly modify the objective function and propose another optimization method, named modification MUSER (mMUSER), to fix the problem. In the proposed method, the orthogonal constraint is promised by solving a LDA-like problem. Experiments are repeated to show the superiority of MUSER, and indicate that mMUSER does improve the performance in some special cases.

## 2 Partial multi-label learning via multisubspace representation

### 2.1 Notation

In the following section, the instance-feature matrix for $n$ samples is denoted by $X = [x_1, x_2, ..., x_n] \in \mathbb{R}^{d\times n}$. The label matrix is arranged by $Y = [y_1; y_2; ...y_n] \in 0, 1^{n\times q}$, where $y_{i,j} = 1$ means the $i$-th sample belongs to $j$-th class.

### 2.2 Formulation

Inspired by [3], the ground truth matrix, denoted by $\widetilde{Y} \in \{0, 1\}^{n\times q}$, can also be assumed to be low-rank in PML. Therefore, $\widetilde{Y}$ is decomposed by a low-dimensional label subspace $U \in \mathbb{R}^{n\times c}$ and the label correlation matrix $P \in \mathbb{R}^{c\times q}$.

$$\widetilde{Y} \backsimeq UP \tag{1}$$

Each original label may be affected by all c latent labels, which implies high-order one-to-all label correlation.

To learn $\widetilde{Y}$ effectively, we minimize the reconstruction error between the candidate label matrix $Y$ and the product of $U$ and $P$ as follows:

$$\min_{U,P} \frac{1}{2}\|Y - UP\|_F^2 + \mathcal{R}(U, P) \tag{2}$$

where $\mathcal{R}(U, P)$ denotes the regularization to control the model complexity.

The ideal latent label subspace is expected to be consistent with intrinsic structural among features [7]. Therefore, a graph Laplacian regularization is introduced to ensure such consistency between features and latent labels. We define the pairwise similarity matrix $S \in \mathbb{R}^{n\times n}$, where $S_{ij} = exp(-||x_i - x_j||_2^2/\sigma^2)$ if the $i$-th instance and $j$-th instance are the $k$-nearest neighbours, otherwise $S_{ij} = 0$. Then the graph regularization term is

$$\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} S_{ij}||\frac{u_i}{\sqrt{E_{ii}}} - \frac{u_j}{\sqrt{E_{jj}}}||_2^2 = Tr(U^T LU) \tag{3}$$

where $L = E^{-\frac{1}{2}}(E-S)E^{-\frac{1}{2}}$ is a graph Laplacian matrix and $E$ is a diagonal matrix with $E_{ii} = \sum_{j=1}^{n} S_{ij}$.

In the real-world application, feature information is often corrupted by outliers and noise. Therefore, a feature correlation matrix $Q \in \mathbb{R}^{m\times c}$ is introduced to map the original feature space to a low-dimensional feature subspace, which can provide compact and discriminative feature information for reducing the negative effects caused by noisy feature information. Here $m$ is the dimension of feature subspace. Therefore, the formulation of MUSER is

$$\begin{aligned} &\min_{W,Q,U,P} \frac{1}{2}||U - X^T QW||_F^2 + \frac{\alpha}{2}||Y - UP||_F^2 + \frac{\beta}{2}Tr(U^T LU) + \mathcal{R}(W, U, P) \\ &s.t. \quad Q^T Q = I \end{aligned} \tag{4}$$

where $\mathcal{R}(W, U, P) = \frac{\gamma}{2}(||W||_F^2 + ||U||_F^2 + ||P||_F^2)$, and the orthogonality constraint for $Q$ is to ensure the latent feature subspace be more compact after mapping.

In summary, MUSER utilizes both label and feature subspace representations to train the desired model, which reduce the negative effects from redundant labels and the feature noise. Combining above two subspaces, the trained PML model is desired to be more effective and robust to both feature and label noises.

In the predict stage, we firstly use $Q$ to map the data into the latent feature subspace, and then use the coefficient matrix $W$ to predict the latent semantics in label space, and finally use label correlation matrix $P$ to recover the ground-truth labels from the label space.

$$\hat{Y} = X^{*T} QWP \tag{5}$$

where $\hat{Y}$ is the prediction label matrix corresponding to the $X^*$.

### 2.3 Optimization of MUSER

The model proposed is convex, which can be solved effectively by alternating optimization scheme.

**Step 1: Calculate** $P$**.** With $U, Q, W$ fixed, Eq.(4) can be reduced to:

$$\min_{p} \frac{\alpha}{2}||Y - UP||_F^2 + \frac{\gamma}{2}||P||_F^2 \tag{6}$$

and we can get the closed form solution:

$$P = (\alpha U^T U + \gamma I)^{-1}\alpha U^T Y \tag{7}$$

**Step 2: Calculate** $U$**.** With $P, Q, W$ fixed, Eq.(4) can be reduced to:

$$\min_{U} \frac{1}{2}||U - X^T QW||_F^2 + \frac{\alpha}{2}||Y - UP||_F^2 + \frac{\beta}{2}Tr(U^T LU) + \frac{\gamma}{2}||U||_F^2 \tag{8}$$

The objective function is differentiable, thus $U$ can be optimized via the standard gradient descent algorithm:

$$\nabla_U = (1+\gamma)U + \beta LU + \alpha UPP^T - \alpha YP^T - X^TQW \tag{9}$$

$$U := U - \lambda_U \nabla_U \tag{10}$$

$\nabla_U$ is the gradient of Eq.(9), $\lambda_U$ is the stepsize of gradient descent

**Algorithm 1** MUSER

```
Input: $X \in \mathbb{R}^d, Y \in \{0,1\}^{n \times q}, \alpha, \beta, \gamma, T_{max}$.
Output: $W, Q, U, P$.
Initialize $W, Q, U, P$ randomly, $t = 1$, $convergence = false$.
while $t < T_{max}$ or $!convergence$ do
    Use Eq.(7) to update $P$
    Use Eq.(10) to update $U$
    Use Eq.(12) to update $Q$
    Use Eq.(15) to update $W$
    $t = t + 1$
    if objective function (4) is convergenced then
        $convergence = true$
    end if
end while
```

**Step 3: Calculate $Q$.** With $P, U, W$ fixed, Eq.(4) can be reduced to:

$$\begin{aligned} &\min_Q \frac{1}{2}||U - X^TQW||_F^2 \\ &s.t. Q^TQ = I \end{aligned} \tag{11}$$

Similarity to **Step 2**, we can get $Q$ as follows:

$$Q := Q - \lambda_Q(-XUW^T + XX^TQWW^T) \tag{12}$$

To satisfy the constraint $Q^TQ = I$, we map each row of $Q$ onto the unit norm ball after each iteration:

$$Q_{i,:} \leftarrow \frac{Q_{i,:}}{||Q_{i,:}||} \tag{13}$$

where $Q_{i,:}$ is the i-th row of $Q$.

**Step 4: Calculate $W$.** With $P, U, Q$ fixed, Eq.(4) can be reduced to:

$$\min_W \frac{1}{2}||U - X^TQW||_F^2 + \frac{\gamma}{2}||W||_F^2 \tag{14}$$

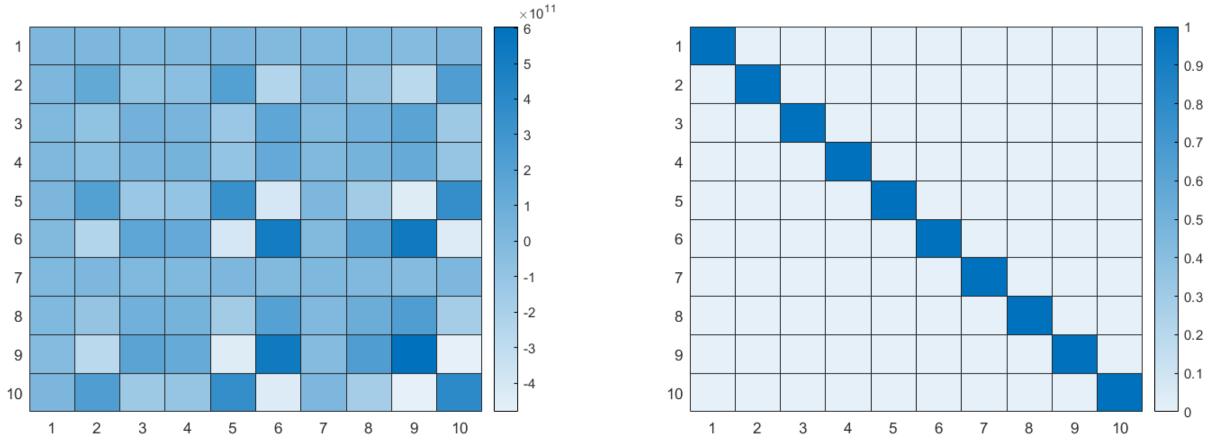
and we can get the closed form solution:

$$W = (Q^TXX^TQ + \gamma I)^{-1}Q^TXU \tag{15}$$

During the entire process of optimization, we first initialize the required variables, then repeat the above steps until the function converges or reach the maximum iterations, which is summarized in Algorithm 1.

### 2.4 A modification for MUSER and its optimization

In fact the Eq.(13) in **Step 3** can NOT ensure the orthogonality of $Q^TQ = I$. We can perform the algorithm and find that normalizing $Q$ will NOT statisfy the orthogonal constraint (Figure.2).



(a) $Q^TQ$ of MUSER

(b) $Q^TQ$ of mMUSER

Figure 2: The matrix of $Q^TQ$ on bibtex dataset of (a) MUSER and (b) mMUSER

To adress this problem, we change the update order of variables. Furthermore, we modify the model by eliminating the regularization term of $W$. The reason for this is that the term $||U - X^TQW||_F^2$ is constrained by $Q^Q = I$ already. The regularization of $W$ may be redundant and degrade the performance of building the correlation between latent label subspace and latent feature subspace. Therefore, we proposed a modification of MUSER (Eq.(4)) named mMUSER:

$$\begin{aligned} &\min_{W,Q,U,P} \frac{1}{2}||U - X^TQW||_F^2 + \frac{\alpha}{2}||Y - UP||_F^2 + \frac{\beta}{2}Tr(U^TLU) + \mathcal{R}(U,P) \\ &s.t. \quad Q^TQ = I \end{aligned} \tag{16}$$

### 2.5 Optimization of mMUSER

**Step 1: Calculate $P$.** This step will be the same as optimization of MUSER in previous analysis.

The **Step 2**, **Step 3** and **Step 4** will be different. It is expected to transfer the problem with orthogonal constraint into a eigenvalue problem, which means it should be solved in the last step. As a result, we update $W$ before $Q$.

**Step 2: Calculate $U$.** We can still use the Gradient Descent to update $U$. However, it is worth mentioned that the update formulation of $U$ can be transferred into a sylvester equation. Let $\nabla_U = 0$

$$\nabla_U = [(1+\gamma)I - \beta L]U + \alpha UPP^T - \alpha YP^T - X^TQW = 0 \tag{17}$$

then we have

$$AU + UB = C \tag{18}$$

where $A = [(1+\gamma)I - \beta L]$, $B = PP^T$ and $C = \alpha YP^T + X^TQW$

**Step 3: Calculate $W$.** With $P, U, Q$ fixed, Eq.(16) can be reduced to:

$$\min_W \frac{1}{2}||U - X^TQW||_F^2 \tag{19}$$

and we can get the closed form solution:

$$W = (Q^TXX^TQ)^{-1}Q^TXU \tag{20}$$

**Step 4: Calculate $Q$.** With $P, U, W$ fixed, Eq.(16) can be reduced to:

$$\begin{aligned} &\min_{Q} \frac{1}{2}||U - X^T QW||_F^2 \\ &s.t. Q^T Q = I \end{aligned} \tag{21}$$

By substitute Eq.(20) into Eq.(21), we have

$$\begin{aligned} &\frac{1}{2}||U - X^T QW||_F^2 \\ =&\frac{1}{2}Tr[(U - X^T QW)^T (U - X^T QW)] \\ =&\frac{1}{2}Tr(U^T U - 2U^T X^T QW + W^T Q^T X X^T QW) \\ =&\frac{1}{2}Tr[U^T U - (Q^T X X^T Q)^{-1} Q^T X U U^T X^T Q] \end{aligned} \tag{22}$$

Therefore, the optimization problem of $Q$ is transferred into

$$\begin{aligned} &\max_{Q} Tr[(Q^T X X^T Q)^{-1} Q^T X U U^T X^T Q] \\ &s.t. Q^T Q = I \end{aligned} \tag{23}$$

It is easy to see that the problem (23) is an **orthogonal LDA-like problem**, where $S_t = XX^T$ is the total scatter matrix and $S_w = XUU^T X^T$ is the within-class.

**Algorithm 2** mMUSER

```
Input: $X \in \mathbb{R}^d, Y \in \{0,1\}^{n \times q}, \alpha, \beta, \gamma, T_{max}$.
Output: $W, Q, U, P$.
Initialize $W, Q, U, P$ randomly, $t = 1$, $convergence = false$.
while $t < T_{max}$ or $!convergence$ do
    Use Eq.(7) to update $P$
    Use Eq.(10) to update $U$
    Use Eq.(20) to update $W$
    Use Eq.(23) to update $Q$
    $t = t + 1$
    if objective function (4) is convergenced then
        $convergence = true$
    end if
end while
```

# 3 Experiment

## 3.1 Experiment set up

We conduct experiments on three PML datasets, which are synthesized from LIBSVM database including Bibtext, delicious and BlogCatalog [8]. These datasets are added with redundant noise labels by the controlling parameter $r$. Here, $r \in \{1, 2, 3\}$ represents the average number of false positive labels for training examples. Furthermore, we choose the dimension of label subspace $c$ as $50\%q$ and feature subspace $m$ as $50\%d$, and set the learning rate of $\lambda_U, \lambda_Q$ to be 0.2. Table 1 shows the characteristics of the experimental datasets.

We also set the trade-off parameters according to the suggestions in respective literatures. Parameters in MUSER method including $\alpha, \beta, \gamma$ are chosen from $\{10^{-2}, 10^0, 10^2\}$ with a grid search manner.

Five widely-used multi-label metrics are employed to evaluate each comparing method, including Hamming Loss, Ranking Loss, One-Error, Coverage and Average Precision.

Table 1: Characteristics of the employed experimental datasets. For each dataset, the number of examples (#$n$), the dimension of features (#$d$), and the number of class labels (#$q$).

| Dataset | #n | #d | #q |
|---|---|---|---|
| Bibtex | 7359 | 1836 | 159 |
| delicious | 16105 | 500 | 983 |
| BlogCatalog | 10312 | 128 | 39 |

### 3.2 Experiment results and discussions

In this subsection, we repeat the experiment by our code on three different datasets. The results are shown in the following tables and the discussions are also given.

Table 2: Comparison of MUSER(from original paper's experiment data), MUSER(from ) and mMUSER on bibtex dataset with five evaluation metrics, where the direction of arrow points to represents the better and the best performances are shown in bold face.

| $r = 1$ (One redundant label for each instance) | | | | | |
|---|---|---|---|---|---|
| | **Hamming↓** | **Ranking↓** | **One-Error↓** | **Coverage↓** | **Average Precision↑** |
| **MUSER(O)** | **0.0090** | 0.1230 | 0.3770 | **0.2310** | 0.5500 |
| **MUSER** | 0.9849 | **0.0492** | **0.3262** | 14.8798 | 0.3452 |
| **mMUSER** | 0.9849 | 0.3025 | 0.7812 | 66.4154 | **0.6683** |

Table 3: Comparison of MUSER and mMUSER on delicious dataset with five evaluation metrics, where the direction of arrow points to represents the better and the best performances are shown in bold face.

| $r = 1$ (One redundant label for each instance) | | | | | |
|---|---|---|---|---|---|
| | **Hamming↓** | **Ranking↓** | **One-Error↓** | **Coverage↓** | **Average Precision↑** |
| **MUSER** | **0.9721** | **0.1941** | **0.4329** | **642.9043** | 0.1816 |
| **mMUSER** | 0.9785 | 0.2246 | 0.6129 | 670.8687 | **0.2175** |

- Table 2 show the performance of MUSER (from original dpaer), MUSER (from this report) and mMUSER on bibtex dataset. On one hand, it is shown that the results from this report often perform better. The results from original data is perform by ten-fold cross-validation, while the result in this report did not use this method for simpilicity. In fact, they are closed to each other. As a result, the results from this report is equal to the result from original paper in probability. On the other hand, mMUSER is not as good as expected, which only wins on the metric of Average Precision.

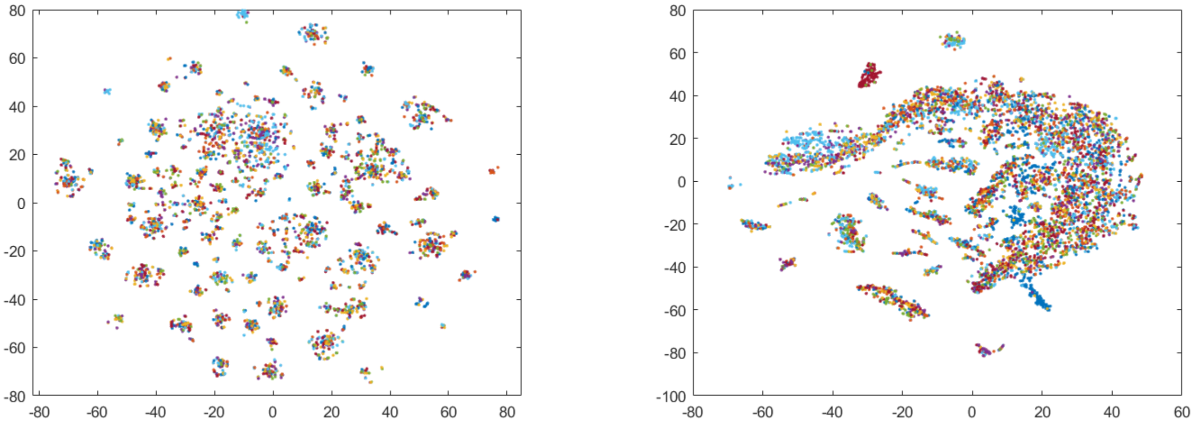
- Two kinds of experiment results are different with original paper. First, the Hamming loss are counted by obtained by the XOR operation of the ground labels and the predicted labels. However, the predicted labels are given by scores in MUSER, which can not be measured by Hamming loss. The detail of computation of Hamming loss is not mentioned in detail in the original paper, but we still give the Hamming loss computed by its definition as experiment results. Second, the metric of coverage may be not compute by its definition, which should be the minimum length

Table 4: Comparison of MUSER and mMUSER on BlogCatalog dataset with five evaluation metrics, where the direction of arrow points to represents the better and the best performances are shown in bold face.

| | Hamming↓ | Ranking↓ | One-Error↓ | Coverage↓ | Average Precision↑ |
|---|---|---|---|---|---|
| $r = 1$ (One redundant label for each instance) | | | | | |
| **MUSER** | 0.2591 | **0.0672** | **0.1673** | **4.1971** | **0.1568** |
| **mMUSER** | 0.2591 | 0.0682 | 0.2132 | 4.2494 | 0.1473 |
| $r = 2$ (Two redundant label for each instance) | | | | | |
| | **Hamming↓** | **Ranking↓** | **One-Error↓** | **Coverage↓** | **Average Precision↑** |
| **MUSER** | 0.2591 | **0.0672** | 0.1804 | 4.2266 | 0.1518 |
| **mMUSER** | 0.2591 | 0.0681 | **0.1591** | **4.2249** | **0.1732** |
| $r = 3$ (Three redundant label for each instance) | | | | | |
| | **Hamming↓** | **Ranking↓** | **One-Error↓** | **Coverage↓** | **Average Precision↑** |
| **MUSER** | 0.2591 | **0.0685** | 0.1912 | 4.2889 | **0.1390** |
| **mMUSER** | 0.2591 | 0.0687 | **0.1774** | **4.2850** | 0.1251 |

covering all ground truth labels in predicted scores ranked.

- Table 3 and 4 show the performance on delicious dataset and BlogCatalog dataset, which are not used in the original paper. The experiment results not only show the superiority of MUSER, but also prove that mMUSER does improve the model in some special case.
- The clustring (computed by $X^T Q$) in MUSER performs better than that in mMUSER, which is shown in Figure 3 and 4. It is obvious that the data points are clearly grouped into clusters in MUSER, while the margins between them are not evident in mMUSER. Furthermore, the visualization of predict ground truth label approximating $U$ in Figure 5 shows that the clustring in mMUSER is excellent in this perspective though it is not as good as MUSER.



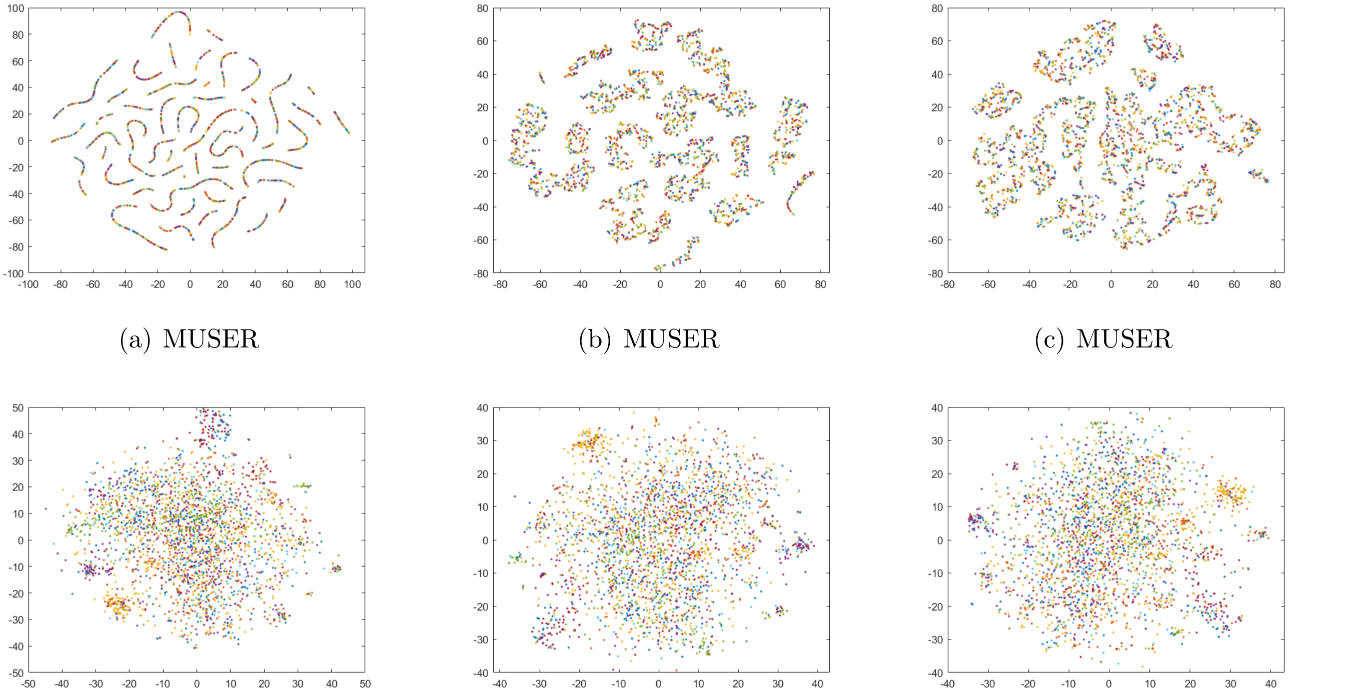
(a) MUSER

(b) mMUSER

Figure 3: The t-sne dimensionality reduction visualization on bibtex dataset of (a) MUSER and (b) mMUSER

## 4 Conclusion

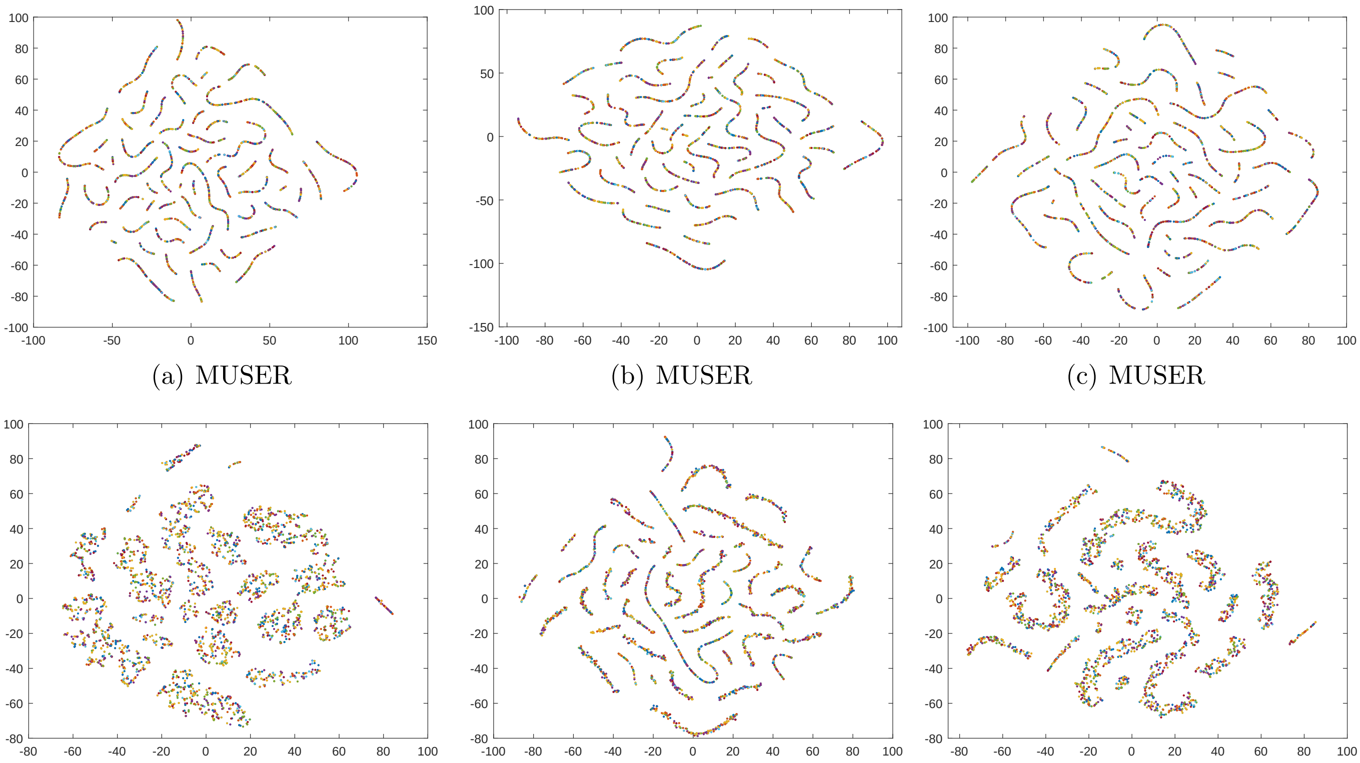
In this report, we introduced the MUSER model in detail, which trains a robust model by considering the noise in both feature space and label space. In particular, MUSER use low-rank decomposition to reduce the negative effects of redundant labels and introduce graph Laplacian regularization to



(a) MUSER (b) MUSER (c) MUSER

(d) mMUSER (e) mMUSER (f) mMUSER

Figure 4: The t-sne dimensionality reduction visualization of $X^TQ$ on BlogCatalog dataset of (a) MUSER and (b) mMUSER



(a) MUSER (b) MUSER (c) MUSER

(d) mMUSER (e) mMUSER (f) mMUSER

Figure 5: The t-sne dimensionality reduction visualization of $X^TQW$ on BlogCatalog dataset of (a) MUSER and (b) mMUSER

ensure the label subspace be in consistent with features, then we utilize feature subspace mapping and orthogonal subspace projection to provide a discriminative feature information. However, the optimization of MUSER can not satisfy the orthogonal constraint of projection matrix. To adress this problem, we modify the model and proposed the mMUSER. Experiments conduct on three PML datasets show the superiority of MUSER, and mMUSER does improve the performance to some extend.

## References

[1] M.-L. Zhang and Z.-H. Zhou, “A review on multi-label learning algorithms,” *IEEE transactions on knowledge and data engineering*, vol. 26, no. 8, pp. 1819–1837, 2013.

[2] M.-K. Xie and S.-J. Huang, “Partial multi-label learning,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, 2018.

[3] G. Yu, X. Chen, C. Domeniconi, J. Wang, Z. Li, Z. Zhang, and X. Wu, “Feature-induced partial multi-label learning,” in *2018 IEEE International Conference on Data Mining (ICDM)*, pp. 1398–1403, IEEE, 2018.

[4] L. Sun, S. Feng, T. Wang, C. Lang, and Y. Jin, “Partial multi-label learning by low-rank and sparse decomposition,” in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, pp. 5016–5023, 2019.

[5] M.-L. Zhang and J.-P. Fang, “Partial multi-label learning via credible label elicitation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 10, pp. 3587–3599, 2020.

[6] H. Wang, W. Liu, Y. Zhao, C. Zhang, T. Hu, and G. Chen, “Discriminative and correlative partial multi-label learning.,” in *IJCAI*, pp. 3691–3697, 2019.

[7] Y. Zhu, J. T. Kwok, and Z.-H. Zhou, “Multi-label learning with global and local label correlation,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 6, pp. 1081–1094, 2018.

[8] C.-C. Chang and C.-J. Lin, “LIBSVM: A library for support vector machines,” *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.