Research Proposal on Partial Mutli-Label Learning

An approach to solve the PML problems in long-tailed distribution dataset

ShengYe Wu
ShanghaiTech University*

Contents

*wushy@shanghaitech.edu.cn

1 Introduction and Motivation

Multi-label learning(MLL) deals with the problem where each object is assigned with multiple class labels simultaneously. For example, an image may be annotated with labels *sea*, *sunset*, *beach*. The task of multi-label learning is to train a classification model that can predict all the relevant labels for unseen instances.

Usually, a common assumption is adopted that each training instance has been precisely annotated with all of its relevant labels, which means the observed label set should be the ground-truth. However, it is difficult to get the precise annotation because the annotators may be confused with some similar labels and ambiguity is caused. Therefore, the label set may contain ground-truth label and some irrelevant labels, which makes the learning task more challenging.

We formalize this learning problem as a new framework called partial multi-label learning(PML). More specifically, PML tries to learn a multi-label model from partially labeled training examples, where each instance is annotated with a set of candidate labels, indicating the following supervised information: a) the candidate set may consist of both relevant and irrelevant labels; b) the number of relevant labels in the candidate set is at least one but unknown; c) labels not in the candidate set are irrelevant to the instance.[1]

For instance, in crowdsourcing image tagging (Figure 1), among the set of candidate labels given by crowdsourcing annotators only some of them are valid ones due to potential unreliable annotators. The task of partial multi-label learning is to learn a multi-label predictor from PML training examples which can assign a set of proper labels for the unseen instance.

Figure 1: An examplar partial multi-label learning scenario. In crowdsourcing image tagging, among the set of 7 candidate labels given by crowdscourcing annotators, only 4 of them are valid ones including house, tree, lavender and France.[7]

Dealing with PML problems is very important. Mostly, the observed label sets contain some irrelevant labels, so adopting traditional MLL algorithms directly will not perform well. Besides, it will be a great cost if we want to make the label sets precise. Thus, developing algorithms to deal with PML problems is meaningful.

2 Literature Reviewing

Previously, there is some methods to deal with PML problems. Xie and Huang propose PML-fp and PML-lc algorithm, which optimizes the label confidence values and trains the model by minimizing the ranking loss.[1] Feng and Sun assume that the label matrix can be composed into two matrices: one is a low-rank ground-truth label matrix, and the other is a sparse noise matrix.[2] On the basis of [2], Xie and Huang divide the Predict matrix W into two parts: label classifier U and noisy label identifier V.[3] The work in [4] uses Low-Rank Representation (LRR) [5] and make global label corelations and local label correlations

respectively. Li and Lyu propose that the noise in feature matrix should also be considered, and they reduce the noise by the means of projection.[6] Fang and Zhang propose PML-VLS and PML-MAP, which elicit credible labels and learns model induction from the candidate label set via an iterative label propagation procedure.[7] Yan and Li propose a self-ensemble approach to deal with PML problems.[8]

To deal with PML problem, an obvious idea is to apply MML algorithms after matrix processing. A common assumption is that the ambiguity exists when there are two similar labels and the annotator can't make a precise decision. Therefore, the noise in the label set is sparse. According to the definition of PML, the rest of label set is ground-truth, and it should be low-rank because of the similarity of the labels.(e.g. In [2] the label set Y is decomposed to P and Q. P is a low-rank matrix and Q is a sparse matrix.) Then we can use MLL algorithms to deal with the low-rank matrix.

However, these kinds of methods also have disadvantages. In the real world, it is difficult to ensure that the data set is like what it is expected to be.(e.g. long-tailed distributions). The low-rank assumption will not perform well under these situations. In this research proposal, we will use Focol-Loss function to deal with PML problems when the dataset is long-tailed distributed.[11]

3 Related Work

3.1 Partial Multi-Label Learning by Low-Rank and Sparse Decomposition

Given the noise-corrupted label matrix, how to identify the ground-truth labels of the instance from the candidate label set and how to train an efficient and robust multi-label classifier for label prediction are two challenging problems in PML. In this paper, a PML-LRS method that acquires the accurate label matrix using the concept of low-rank and sparse decomposition is proposed, thereby predicting the labels of unlabeled data more accurately.[2]

$$\min_{P,Q,W} \frac{\eta}{2} ||P - WX||_F^2 + ||P||_* + \beta ||Q||_1 + \gamma ||W||_*$$

$$s.t. Y = P + Q$$

3.2 Focal Loss

The Focal Loss is designed to address the one-stage object detection scenario in which there is an extreme imbalance between foreground and background classes during training. Focal loss places a higher weight on "hard to classify" instances so that the effect of the disbalanced distribution on it will be lower.By multiplying a modulating factor to BCE (with the tunable focusing parameter $\theta \geq 0$), Focal-Loss has the form as follows:[11]

$$L_{FL} = \begin{cases} -\left(1 - p_i^k\right)^\theta log\left(p_i^k\right), if\ y_i^k = 1 \\ -\left(p_i^k\right)^\theta log\left(1 - p_i^k\right), otherwise \end{cases}$$

We set $\theta = 2$ according to [11] for convenience.

4 Proposed Methods

4.1 The Framework

Our methods is to take two matrices as input: the instance matrix $X = [x_1, x_2, \cdots, x_n] \in R^{k \times n}$, where k is the dimension of the feature vector and n is the number of training instances. And we define $Y = [y_1, y_2, \cdots, y_c]^T \in \{0, 1\}^{c \times n}$, to represent the label assignments for the corresponding labeled examples, where c is the number of labels. The values in this matrix are within $\{0, 1\}$. Then we define a classifier $W \in R^{c \times k}$, which enables $Y = WX$. Divide the classifier W into three parts: $A \in R^{c \times k}$, which corresponds to the head part of the dataset; $B \in R^{c \times k}$, which corresponds to the tail part of the dataset; $C \in R^{c \times k}$, which corresponds to the partial part of the dataset. (Partial part is those labels which are ambigious.) The relationship of those three parts of data is shown in the following table.

	Head	Tail	Partial
Ground truth	✓	✓	×
Label correlation	✓	×	✓
Supported by the features	✓	✓	✓
Low rank	✓	×	?
Sparse	×	✓	✓

Whether the C is low rank or not is uncertain in this table. However, according to the assumption, we can know that CX, which means the partial part, should be sparse.

Adopt Focal-Loss function as the loss function, we propose a PML-fl method to learn the multi-label learning model by solving the following problem:

$$\min_{A,B,C} L_{FL}\left(X, Y, A, B, C\right)$$

In the Focal-Loss function, we use sigmoid function to compute p_i^j, $p_i^j = \sigma\left((A_j + B_j + C_j) \cdot x_i\right)$.

According to the definition of PML problems, the ambiguity will be generated because there is strong label correlation between the ambigious labels and the ground truth. Therefore,in a long-tailed distributed dataset, the ambigious labels should be added into the head part since the head part have a strong label correlation with it. And for the tail part, since the labels in it don't have strong label correlation with those in head part, they also don't have strong label correlation with the partial labels. Together with the table, we can know that: B and CX are sparse matrices, and A+C is a low-rank matrix. Therefore the above optimization problem becomes:

$$\min_{A,B,C} L_{FL}\left(X, Y, A, B, C\right) + \alpha \left|\left|B\right|\right|_0 + \beta \left|\left|CX\right|\right|_0 + \gamma rank\left(A + C\right)$$

This optimization problem is difficult to solve because the rank and cardinality operators are discontinuous and non-convex. Therefore, these operators are respectively relaxed to their convex surrogates: the nuclear norm[12] and the l_1-norm[13]. The final objective function can be formulated as follows:

$$\min_{A,B,C} L_{FL}\left(X, Y, A, B, C\right) + \alpha \left|\left|B\right|\right|_1 + \beta \left|\left|CX\right|\right|_1 + \gamma \left|\left|A + C\right|\right|_*$$

Finally, after we get the optimal solution A^*, B^*, we can get the prediction: $\hat{Y} = (A^* + B^*) \cdot X$.

4.2 Optimization

To solve this problem, first we convert it into the following equivalent problem.

$$\min_{A,B,C} L_{FL}\left(X, Y, A, B, C\right) + \alpha \left|\left|B\right|\right|_1 + \beta \left|\left|E\right|\right|_1 + \gamma \left|\left|D\right|\right|_*, \ s.t.\ A + C = D, CX = E$$

Covert it into an augmented Lagrange function:

$$\min_{A,B,C,D,E} L_{FL}\left(X, Y, A, B, C\right) + \alpha \left|\left|B\right|\right|_1 + \beta \left|\left|E\right|\right|_1 + \gamma \left|\left|D\right|\right|_* + Q_1^T\left(A + C - D\right) + \frac{\mu_1}{2}\left|\left|A + C - D\right|\right|_F^2 + Q_2^T\left(CX - E\right) + \frac{\mu_2}{2}\left|\left|CX - E\right|\right|_F^2$$

where $Q_1 \in R^{c \times k}, Q_2 \in R^{c \times n}$, μ_1, μ_2 is a penalty parameter. It can be rewritten as:

$$\min_{A,B,C,D,E} L_{FL}\left(X, Y, A, B, C\right) + \alpha \left|\left|B\right|\right|_1 + \beta \left|\left|E\right|\right|_1 + \gamma \left|\left|D\right|\right|_* + \frac{\mu_1}{2}\left|\left|A + C - D + \frac{Q_1}{\mu_1}\right|\right|_F^2 - \frac{\mu_1}{2}\left|\left|\frac{Q_1}{\mu_1}\right|\right|_F^2 + \frac{\mu_2}{2}\left|\left|CX - E + \frac{Q_2}{\mu_2}\right|\right|_F^2 - \frac{\mu_2}{2}\left|\left|\frac{Q_2}{\mu_2}\right|\right|_F^2$$

1. When keeping A,B,C,E fixed, optimizing the function is equivalent to the following problem:

$$\min_D \gamma \left|\left|D\right|\right|_* + \frac{\mu_1}{2}\left|\left|A + C - D + \frac{Q_1}{\mu_1}\right|\right|_F^2$$

which is equivalent to:

$$\min_D \frac{\gamma}{\mu_1} ||D||_* + \frac{1}{2} \left|\left|D - \left(A + C + \frac{Q_1}{\mu_1}\right)\right|\right|_F^2$$

Adopt singular value decomposition(SVD) to $A + C + \frac{Q_1}{\mu_1}$, where $A + C + \frac{Q_1}{\mu_1} = U \cdot S \cdot V^T$. Apply some soft thresholding on the singular values, we can get the solution, where S is the soft thresholding function:

$$D = U \cdot S_{\frac{\gamma}{\mu_1}}[S] \cdot V^T$$

2. With B,C,D,E fixed, we obtain the following equation for A:

$$\min_A L_{FL}(X, Y, A, B, C) + \frac{\mu_1}{2} \left|\left|A + C - D + \frac{Q_1}{\mu_1}\right|\right|_F^2$$

This is a logistic regression problem. It is easy to get the solution for matrix A. We will show the process of getting the partial derivative of Focal Loss function:

$$p = \frac{1}{1+e^{-\theta * X}}, \frac{\partial p}{\partial \theta} = p(1-p)X$$

$$L_{FL} = \frac{1}{n} \sum -ylog(p)(1-p)^2 - (1-y)log(1-p)p^2$$

$$\begin{aligned}\frac{\partial L}{\partial \theta} = \frac{1}{n} \Big[&-y(1-p)^3 + 2ylog(p)p(1-p)^2 + (1-y)p^3 \\ &- 2(1-y)log(1-p)p^2(1-p)\Big] X\end{aligned}$$

3. With A,C,D,E fixed, solve the problem for B by the following function:

$$\min_B L_{FL}(X, Y, A, B, C) + \alpha ||B||_1$$

This is a logistic regression problem which contains L1 regularization. We can adopt proximal gradient descent to solve it. Set $h(B) = L_{FL}(X, Y, A, B, C)$ and the iteration for B is:

$$B^{k+1} = S_{\alpha t}\left(B^k - t \cdot \nabla h(B)\right)$$

4. Fix A,B,D,E and solve the following problem for C:

$$\min_C L_{FL}(X, Y, A, B, C) + \frac{\mu_1}{2} \left|\left|A + C - D + \frac{Q_1}{\mu_1}\right|\right|_F^2 + \frac{\mu_2}{2} \left|\left|CX - E + \frac{Q_2}{\mu_2}\right|\right|_F^2$$

This is a logistic regression problem. It is easy to get the solution for matrix C. The process is similar to that of A.

5. Fix A,B,C,D and solve the following problem for E:

$$\min_E \beta ||E||_1 + \frac{\mu_2}{2} \left|\left|CX - E + \frac{Q_2}{\mu_2}\right|\right|_F^2$$

This is a Lasso problem and we can adopt proximal gradient descent to solve it. Set $f = CX + \frac{Q_2}{\mu_2}$. Since the coefficent matrix of E is an identity matrix, we can get the optimal solution:

$$E = S_{\frac{\beta}{\mu_2}}(f)$$

6. Finally, the Lagrange multiplier matrix Q_1, Q_2 and regularization term μ_1, μ_2 are updated:

$$Q_1^{k+1} = Q_1^k + \mu_1^{k+1}(A + C - D),\ \mu_1^{k+1} = \rho \mu_1^k$$

$$Q_2^{k+1} = Q_2^k + \mu_2^{k+1}(CX - E),\ \mu_2^{k+1} = \rho \mu_2^k$$

where ρ is a positive scalar.

Algorithm 1 PML-fl

Inputs:

X,Y: The partial multi-label training set;

α, β, γ**: the learning parameter;**

Process:

1. Initialize the head classifier A, the tail classifier B and the partial classifier C.
2. Initialize the parameter α, β, γ.
3. Initialize the auxiliary matrices D=A+C,E=CX, the Lagrangian multiplier matrices $Q_1 \in R^{c\times k}, Q_2 \in R^{c\times n}$ and hyperparameters μ_1, μ_2.
4.

while not converged **do**

Step1

Update A,C;

Step2

Obtain D by the soft thresholding;

Obtain B and E by solving the Lasso problems;

Update Q_1, Q_2 and μ_1, μ_2;

t=t+1;

end while

Outputs: A^*,B^*,C^*

4.3 Computation Complexity analysis

The time complexity mainly consists of the SVD decomposition and the matrix multiplication. The SVD decomposition needs $O\left(kc^2\right)(k > c)$. The matrix multiplication needs $O\left(kcn\right)$. Suppose the times of iteration is t, then the total time complexity is $O\left(t * \left(kc^2 + kcn\right)\right)$.

5 Stimulation

To verify the effectness of PML-fl algorithm, a stimulation experiments has been designed.

5.1 The intialization of the datasets

First, we initialize a matrix which satisfy the Gaussian distribution N(0,25). We run a SVD decompositon on it and assume the result is matrix A+C. This can ensure that A+C is a low rank matrix. Then we choose 0.6% elements in A randomly as the elements of C, to ensure that C is a sparse matrix. To satisfy that CX is sparse, we initialize a sparse instance set X which satisfies the Gaussian distribution N(0,9), with a density of 10%. For matrix B, we initialize a sparse matrix which satisfies Gaussian distribution N(0,16), with a density of 3%. Meanwhile, we intialize a matrix ϵ which satisfies Gaussian distribution N(0,1) as the white noise.

We get the label matrix $Y = (A + B + C)X + \epsilon$. Put the elements in Y into the sigmoid function, we can get the probability of being positive label in each position. Usually, the number of positive labels should be a bit fewer than that of negative labels. Thus we choose threshold = 0.6 to change matrix Y into a binary label set.

5.2 Hyperparameters setting and the result

We choose $\alpha = 0.08, \beta = 0.05, \gamma = 0.1$ as the combination of hyperparameters. The learning rate of solving matrix A and C is set to 0.005, and the learning rate of solving matrix B is set to 0.001. After the iteration has been completed, the value of the total loss is changed as the figure follows.

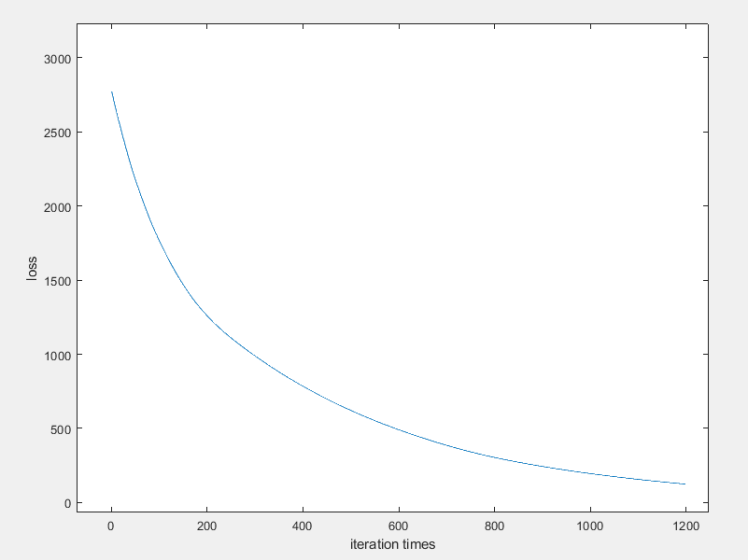

Figure 2: Variation of the total loss function with the number of iterations

According to this figure, the model has converged.

To validate the assumption of PML-fl algorithm we draw the heatmap of matrix B after iteration. From the heatmap of B, we can find that B is sparse, which is the same as the assumption.

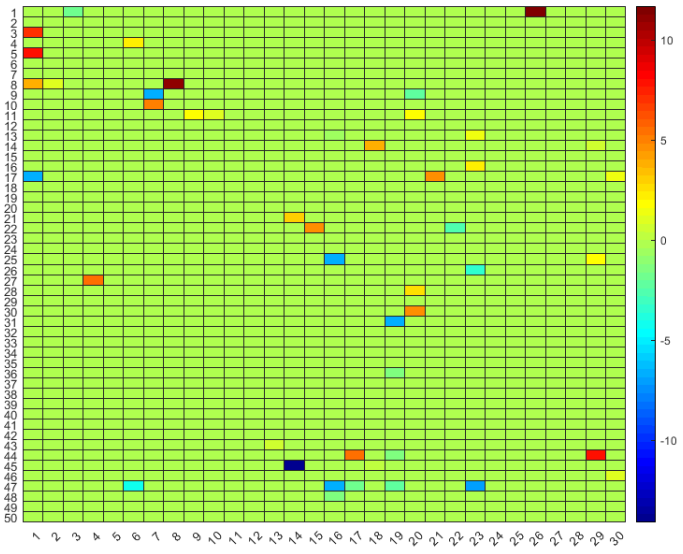

Figure 3: The heatmap of matrix B

Another assumption is that matrix A+C should be low rank. We use the trace norm to approximately solve this non-convex problem. Since the trace norm of A+C has significantly decreased, the assumption is validated.

```
ans = "The inital trace norm: 6712.565732 "
ans = "The trace norm after iteration: 907.649410 "
```

Figure 4: The change of A+C's trace norm

To evaluate the accuracy of the model, we adopt Hamming Loss and F1 score. The Hamming loss is 0.028, and the F1 score is 0.972. So far, through stimulation experiments, we can consider that this algorithm has basically achieved our goal.

6 Reference

[1] Ming-Kun Xie, Sheng-Jun Huang. Partial Multi-Label Learning. In AAAI, pages 4302-4309, 2018.

[2] Lijuan Sun, Songhe Feng, TaoWang, Congyan Lang, Yi Jin. Partial Multi-Label Learning by Low-Rank and Sparse Decomposition. In AAAI, pages 5016-5023, 2019.

[3] Ming-Kun Xie, Sheng-Jun Huang. Partial Multi-Label Learning with Noisy Label Identification. In AAAI, pages 6454-6461, 2020.

[4] Lijuan Sun, Songhe Feng, Jun Liu, Gengyu Lyu, Congyan Lang. Global-Local Label Correlation for Partial Multi-Label Learning. DOI 10.1109/TMM.2021.3055959, IEEE Transactions on Multimedia

[5] Guangcan Liu, Zhouchen Lin, Yong Yu. Robust Subspace Segmentation by Low-Rank Representation. In ICML, 2010.

[6] Ziwei Li, Gengyu Lyu and Songhe Feng. Partial Multi-Label Learning via Multi-Subspace Representation. In IJCAI, pages 2612-2618, 2020.

[7] Jun-Peng Fang, Min-Ling Zhang. Partial Multi-Label Learning via Credible Label Elicitation. In AAAI, pages 3518-3525, 2019.

[8] Yan Yan, ShiNing Li. A Self-Ensemble Approach for Partial Multi-Label Learning.

[9] Jia Zhang , Yidong Lin , Min Jiang , Shaozi Li1 , Yong Tang and Kay Chen Tan. Multi-label Feature Selection via Global Relevance and Redundancy Optimization. In IJCAI, pages 2512-2518, 2020.

[10] Candes, E., and Tao, T. 2005. Decoding by linear programming. arXiv preprint math/0502327.

[11] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, Piotr Dolla. Focal Loss for Dense Object Detection. In ICCV, pages 2980-2988, 2017.

[12] Candes, E. J., and Recht, B. 2009. Exact matrix completion via ‘ convex optimization. Foundations of Computational mathematics 9(6):717.

[13] Candes, E., and Tao, T. 2005. Decoding by linear programming. arXiv preprint math/0502327.