A Consensus-Based Active Learning Strategy for Multi-Label Classification

Prof. Ganesh Ramakrishnan[*]
IITB

Ashish Kulkarni
IITB

Simoni Shah
IITB

Pankaj Singh
IITB

Aditya Kumar Akash
IITB

ABSTRACT

Generating quality labeled data, while essential for automatic classification, is a time consuming task. The effort is further accentuated when the task involves multiple labels and multiple labelers—both human and machine. Here, the quality of labeled data depends on gathering all possible labels for all data points while ensuring a high consensus across labelers. How to minimize human labeling effort without sacrificing the accuracy of individual labelers or inter-labeler consensus for a multi-label classification task is a challenging problem and the focus of this work. While prior work has addressed specific instances of this problem involving either a single label or single labeler, we study the problem in its entirety, motivated by its application to several areas like video tagging, targeted advertising *etc.* We propose a novel formulation that aims to collectively optimize the cost of labeling, labeler reliability and inter-labeler consensus. Our solution leverages the pool-based active learning paradigm and interactively samples instance-label pairs from the unlabeled pool. Specifically, our sampling strategy queries for instance-label pairs that maximize the expected consensus in successive labeling iterations. Experiments on several real-world datasets (publicly available) demonstrate the efficacy of our approach in achieving promising classification results with much fewer labeled data than state-of-the-art methods.

Keywords

Consensus Learning; Active learning; Multi-Label Classification

1. INTRODUCTION

Multi-Label learning such as prediction of labels for videos is an expensive task and has received a significant attention. The problem becomes more difficult when the learning is unsupervised. In such situations, gathering of labels is done by exploiting the predictions from different sources. Combining the predictions from different sources while also maintaining a high consensus amongst the labelers is a desirable solution. A passive combination of prediction of labelers, both machine and human, would only maximize the consensus amongst them. There is no learning element in the system, except for the prediction combination. Thus, we need an active learning formulation, where we query the labelers with selective instances and labels, to improve the current predictions and also provide an option to be labelers to learn from the query about the previously labelled instance.

In multi-label classification problems involving machines and human labelers together, it is desirable that with time the dependency on human labelers reduce and the machine labelers perform further classification with very little help from a group of experts. The expertise of the users needs to also come in such a problem. In this paper, we try to build a system which aims to collectively optimize labeling cost and improving inter-labeler consensus. Another major focus of our work is to show that improving the consensus amongst the users also leads to model prediction getting more aligned towards the ground truth.

2. RELATED WORK

Multi-label active learning has received a significant attention in recent and has successfully been applied to domains such as image annotation and text categorization. Existing multi-label active learning research mainly focus on querying of instances based on minimization of the loss of the classifier and uncertainty sampling. Given the instance the method query for all the labels of the instance, thus wasting labeler's effort since labels are correlated. *Multi-Label active learning* with crowdsourcing has been also considered in past works. These methods query for most uncertain instance, label pair from the most reliable annotator. But none of the previous work try to improve the consensus amongst the labelers while performing the active learning. In this work, we build upon active learning with consensus maximization.

3. PROBLEM DEFINITION AND MODEL

We are given a set $\mathbf{X}$ of instances $\mathbf{x}_1, \ldots, \mathbf{x}_n$ where each $\mathbf{x}_i$ is a feature vector of an instance and a label set L of size $|L| = l$. Then the task of multi-label classification is to assign to each instance $\mathbf{x}_i$ a subset of labels, represented by a binary label vector $\mathbf{z}_i = [z_i^1 \ldots z_i^l]$, where $z_i^j \in \{0, 1\}$. If $\mathcal{Z} =$

[*] BTP Guide

WOODSTOCK '97 El Paso, Texas USA
© 2016 ACM. ISBN 123-4567-24-567/08/06...$15.00
DOI: 10.475/123_4

$\{0, 1\}^l$ denotes the set of all possible label combinations, then the multi-label classification problem can be expressed as a decision function $f : \mathbf{X} \rightarrow \mathcal{Z}$.

Table 1: Notations

Symbol	Meaning
m	Number of multilabel classifiers
n	Number of Instances
l	Number of labels
$\boldsymbol{x}$	An instance
$\|\|.\|\|$	Frobenius Norm
A	$a_{i,j}$ is the prediction of label (j mod l) on instance $\boldsymbol{x}_i$ by model $\lfloor j/l \rfloor$
B	Label node class distribution
U	$u_{i,l}$ is the probability that label l is relevant to $\boldsymbol{x}_i$
Q	$q_{i,l}$ is the probability of seeing label l given label j

We have m labelers, both machine and human. Let us call them models. The predictions of the models are denoted by $\mathbf{Y^k}$, which is a matrix of $\mathbf{n} \times \mathbf{l}$ elements, specifying the presence of label in the instance. For combining the output of each model, we use MLCM-r model outlined in []. The following section briefly talks about MLCM-r.

3.1 MLCM-r

The prediction information of models is encoded in matrix A, which is n by v ($v = m \times l$), where $(i, (k-1) \times l + j)$-th entry is 1 if the k-th model predicts that the j-th label is relevant to the i-th instance, otherwise that entry is set to 0.

We construct a bipartite graph with A as the connection matrix. The graph has n instance nodes and $v = m \times l$ group nodes. A group node represents a label. The group nodes are annotated by letter g and the instance nodes by letter x. An instance node can be connected to more than one group nodes from a single classifier, representing the multilabel predictions of that classifier. A dashed rectangle surrounds group nodes which belong to the same classifier. There is another layer of nodes called label nodes which help represent the label the group node stands for. An example graph has been shown in Figure 3.1 for 2 instances, 3 labels and 2 classifiers.

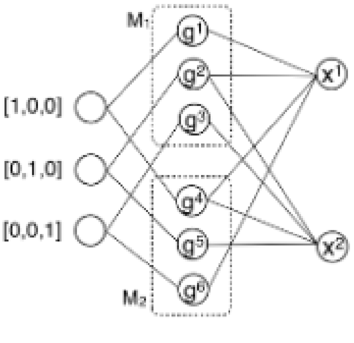

Figure 1: Bipartite Graph for MLCM-r[?]

Each node in the graph has a probability distribution associated with it.

1. Each instance node i is associated with probability distribution $\boldsymbol{u}_i$, where $u_{i,l}$ stands for probability of relevance of l-th label on i-th instance. Matrix $U = [u_1^{'}, ..., u_n^{'}]$ represents this information.

2. Each group node j is associated with probability distribution $\boldsymbol{q}_j$, where $q_{j,l}$ stands for probability of seeing label l given j-th label. This relates to how related two labels are in opinion the the classifier to which the group node j belongs. Matrix $Q = [q_1^{'}, ..., q_2^{'}]$ encodes this information.

3. Each group node j representing label l is also connected to label node which has probability distribution $\boldsymbol{b}_j$, where l-th entry is 1 and others 0. Matrix $B = [b_1^{'}, ..., b_v^{'}]$ encodes this information.

MLCM-r maximizes model consensus by solving the following optimization problem :

$$\min_{U,Q} \sum_{i=1}^{n} \sum_{j=1}^{v} a_{ij} \|\boldsymbol{u}_i - \boldsymbol{q}_j\|^2 + \alpha \sum_{j=1}^{v} \|\boldsymbol{q}_j - \boldsymbol{b}_j\|^2 \tag{1}$$

s.t.

$$u_{ik} \geq 0, \sum_{k=1}^{l} u_{ik} = 1, i = 1, ..., n \tag{2}$$

$$q_{jk} \geq 0, \sum_{k=1}^{l} q_{jk} = 1, j = 1, ..., v \tag{3}$$

In this the first term ensures that if an object $\boldsymbol{x}_i$ is linked to group $\boldsymbol{g}_j$, then their probability distribution should be similar since both of them in some sense represent probability of seeing label while being at that node. The second term ensures that the probability distribution of the groups does not deviate much from its initial distribution. α is the factor of how much we would like to penalize such a constraint violation.

The solution is obtained by block co-ordinate descent :

$$\boldsymbol{q}_j^t = \frac{\sum_{i=1}^{n} a_{ij} \boldsymbol{u}_i^{t-1} + \alpha \boldsymbol{b}_j}{\sum_{i=1}^{n} a_{ij} + \alpha} \tag{4}$$

$$\boldsymbol{q}_j^t = \frac{\sum_{j=1}^{v} a_{ij} \boldsymbol{q}_j^t}{\sum_{j=1}^{v} a_{ij}} \tag{5}$$

Upon convergence, the final probability distributions are given in the rows of U. The main of advantage of MLCM is that it handles the case of label-label correlations present in the data. We use MLCM-r as the consensus model in our system.

3.2 Measure of Consensus

In order to further work with the consensus maximization using active learning, we need to defined a measure of consensus. The consensus of a model with other models is defined using *Cohen Kappa*, which captures inter-rater agreement. It is defined as

$$\kappa = \frac{p_o - p_e}{1 - p_e} = 1 - \frac{1 - p_o}{1 - p_e}, \tag{6}$$

where p_o is the relative observed agreement among raters, and p_e is the hypothetical probability of chance agreement, using the observed data to calculate the probabilities of each observer randomly saying each category. If the raters are in complete agreement then $\kappa = 1$. If there is no agreement among the raters other than what would be expected by chance (as given by p_e), $\kappa \leq 0$.

The consensus of an model is defined as the Kappa value between prediction for the model with the prediction of the MLCM-r model. The overall consensus of the system is the sum of consensus for all the models in the system. The consensus for an instance is taken as mean over Kappa between prediction of MLCM-r and each model.

3.3 Labeler reliability

We also capture labeler reliability in our model. The reliability of a labeler is estimated for each label. For each model, label consider r_i^j as the reliability for user i (we would use user and labeler interchangeably in this paper) for label j. The reliability is updated in following manner

$$r_i^j \leftarrow r_i^j + \gamma(\kappa^j - r_i^j) \tag{7}$$

where κ^j is Kappa calculated w.r.t j^{th} label, and $\gamma < 1$ is a constant.

Accordingly, we update the MLCM-r model. For each group node, instead of a_i^j, we use $a_i^j \times r_i^j$. This accounts for the user reliability in our model.

These form the major component of our model.

4. ACTIVE LEARNING

Based on the above model we learn the consensus prediction, user reliability for labels and consensus of the model. We have two active learning strategy to sample instance, label pairs based on

1. **Uncertainty** based sampling
2. **Influence** based sampling

The difference between these two are present in following subsections.

4.1 Uncertainty based sampling

The underlying assumption for this sampling is that for instance in while the consensus is low indicates that there is confusion amongst the users for deciding the labels for this instance. Thus, the instances with the minimum κ value are the most confused instances.

$$\arg\ \min_x \kappa_x \tag{8}$$

For confused (uncertain) label selection we use following approach. The output of the MLCM model is $\mathbf{u_x}$, a probability distribution across the labels. To obtain a binary label prediction from it, we need to find a threshold τ, such that $l_j = 1$, if $u_j > \tau$, 0 otherwise. A threshold which maximizes the consensus of the instance prediction is selected. The labels which have probabilities of relevance near the threshold τ are the once which are more uncertain. So for a δ interval near τ is used for selecting the confused labels from the confused instance.

The confused instance, label pair is queried to the user i with highest reliability r_i^j on the given label j. This forms the uncertainty based sampling of the instance-label pair.

4.2 Influence based Sampling

This is the most important sampling strategy. In this we sample an instance,label and user set, such that when user is queried about the selected instance, label pair the expected increase in consensus of the system is maximum. The measure of consensus being the same as stated earlier.

$$\arg\ \max_{u,x,l} \sum_u \kappa_u \tag{9}$$

Thus, our aim is to improve the consensus amongst the users, which would lead to improvement in classification.

5. EXPERIMENTS

In this section, we show how each component of the model leads to the improvement in the classification.

5.1 Datasets

Following multilabel classification dataset are used in the experiments :

Table 2: Notations

dataset	#instances	#features	#labels
medical	978	1449	45
enron	1672	1001	53

5.2 Experiment Design

Following experiments are performed

1. We show that the ranking loss of MLCM model improves when we incorporate user reliability.
2. For the demonstrating the effect of reducing the expert size, we remove the user with the least reliability and recalculate the consensus using MLCM-r model. We also show change in F-Measure of the classification.
3. We plot the F1 value of the instances with the κ consensus.
4. For the active learning setting, we perform following experiment
   - Find out instances which have consensus values in a given range, which is selected such that the consensus is neither very high nor low. For our case we choose [0.85, 0.90].
   - Find out the models which have lowest consensus on the selected instance.
   - The models selected add the instances into their training set. The consensus model output is used as ground truth.
   - Consensus model is re-run to calculate the predictions.

   The main objective behind this procedure is to simulate the active learning part on model side. Including instances, with consensus output as Ground truth into training sample, would allow models to collaborate with each other and improve consensus.

The models for experiment 1,2 and 3 is generated using following procedure - A model (one vs all SVM) is obtained by randomly shuffling the dataset, followed by 10-fold CV. For each dataset, we train 10 such models. Each experiment is repeated 10 times.

6. RESULTS

Following table shows ranking loss for MLCM vs MLCM for user reliability

Table 3: MLCM vs MLCM with r_i^j

Model	Avg Raking loss
MLCM	0.01722
MLCM with r_i^j	0.01667

We can see that MLCM with user reliability improves the loss function.
For user reduction experiment following is the table showing average change in ranking consensus vs change in f1 measure.

Table 4: User removal effect

Dataset	Avg Change in Consensus	Avg Change in fMeasure
medical	0.0053068	0.0006081

The removal of least consensus user helps in improving the consensus and also aligns the data with ground truth. Following is the scatter plot of κ vs $F1$ measure :

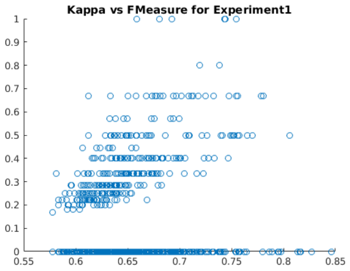

Figure 2: κ vs $F1$ measure for medical data

In the two figures 2 and 3, we can see that the principal component of the plot is along $x = y$ line, which shows that improving the consensus improves the F-Measure.

The results for active learning is in table 5. Clearly the consensus change is positive for both the datasets, which shows the importance of is in table 5.

7. CONCLUSIONS

In this paper, we consider the problem of multi-label classification using active learning. Rather than just sampling the data based on uncertainty we try to sample the instance label pair to maximize the consensus. We provide several evidence to show that the improving the consensus moves the data towards the ground truth. We also incorporate the user-reliability into the system and show the improvement to the original framework proposed. We used this reliability to remove user with least consensus, and show that this led to improved consensus and increased F1 measure.

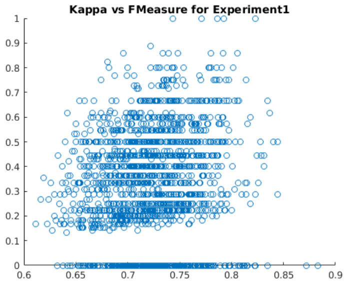

Figure 3: κ vs $F1$ measure for enron data

Table 5: Consensus change by using Active learning

Dataset	Avg Consensus change
Medical	2.22
enron	1.57

8. ACKNOWLEDGMENTS

I would like to thank Prof. Ganesh Ramakrishnan, Ashish Kulkarni, Simoni Shah and Pankaj Singh for providing valuable feedback and insights which helped me complete the work.