Safe Semi-supervised Multi-label Learning

Tong Wei

Oct. 10th, 2016

Introduction

Multi-label learning refers to the problems where an instance can be assigned to more than one category. In this paper, we present a novel Semi-supervised algorithm for Multi-label learning by solving a XXX problem.

Algorithm I

```
Initialize M, the number of learners for each label.
for $t = 1$ to $T$ do
    for $k = 1$ to $K$ do
        Sample M bootstrap replicates
        $\{(\bar{\mathbf{X}}_1, \bar{\mathbf{y}}_1), (\bar{\mathbf{X}}_2, \bar{\mathbf{y}}_2), \ldots, (\bar{\mathbf{X}}_M, \bar{\mathbf{y}}_M)\}$
        for $m = 1$ to $M$ do
            Train an SVM model $\mathcal{M}_{km}$ on $\bar{\mathbf{X}}_m$ and $\bar{\mathbf{y}}_m$.
            Derive $\tilde{\boldsymbol{y}}_{km}$ by predicting on the unlabeled data $X_U$ using
            $\mathcal{M}_{km}$.
        end for
        Compute $\{\mathbf{w}_{k1}, \mathbf{w}_{k2}, \ldots, \mathbf{w}_{kM}\}$ and $\boldsymbol{\mu}_k$ by solving Problem
        (8) in AAAI16.
        Calculate prediction $P_{jk}^t = \sum_{m=1}^{M} \mu_{km} \mathbf{w}_{km}^{\mathrm{T}} \mathbf{x}_j$ for a test data
        $\mathbf{x}_j$.
```

Algorithm II

the number of unlabeled instances

the number of labels

size of $P^{(j)}$ and V are both U × K

```
    end for
end for
Solve $\underset{Y,V}{\operatorname{argmin}} \sum_{i=1}^{\mathrm{U}} \sum_{j=1}^{\mathrm{T}} \left(Y_{ij} - (\mathrm{P}_{i\cdot}^{(j)}) V_{i\cdot}^{\top}\right)^2 + \underbrace{C_1 \sum_{j=1}^{T} \left(\sum_{i=1}^{U} Y_{ij} - q_j\right)^2}_{\texttt{column regulization}} + \underbrace{C_2 \sum_{i=1}^{U} \left(\sum_{j=1}^{T} Y_{ij} - \gamma_0\right)^2}_{\texttt{row regulization}} + C_3 \left|\left|V - V_0\right|\right|_F^2$
```

Algorithm

```
Randomly initialize $\mathbf{Y}$ and $\mathbf{V}$ to real numbers in range (0, 1) and $\sum_{j=1}^{K} V_{ij} = 1$ for all $1 \leq i \leq U$.
Until convergence, do
    Fix $\mathbf{Y}$, update $\mathbf{V}$ using $V_{ik} = V_{ik} - \frac{C3V_{0_{ik}} + \sum_{j=1}^{T} Y_{ij} P_{ik}^{(j)}}{C3 + {}_{j=1}^{T} (P_{ik}^{(j)})^2}$
    Fix $\mathbf{V}$, update $\mathbf{Y}$ using $Y_{ij} = Y_{ij} - \frac{P_{i.}^{(j)} V_{i.}^{\top} + C1 q_j + C2 \gamma_0}{1 + C1 + C2}$
```

Table: Characteristics of the benchmark multi-label data sets.

Data set	#S	dim(S)	L(S)	LCard(S)	LDen(S)	Domain
emotions	593	72	6	1.869	0.311	music
genebase	662	1185	27	1.252	0.046	biology
enron	1702	1001	53	3.378	0.064	text
image	2000	294	5	1.236	0.247	images
scene	2407	294	6	1.074	0.179	images
Yeast	2417	103	14	4.237	0.303	biology
Arts	5000	462	26	1.636	0.063	
Business	5000	438	30	1.588	0.052	
Computers	5000	681	33	1.508	0.046	
Education	5000	550	33	1.461	0.044	
Entertainment	5000	640	21	1.420	0.068	
Health	5000	612	32	1.662	0.052	
Recreation	5000	606	22	1.423	0.065	
Reference	5000	793	33	1.169	0.035	
Science	5000	743	40	1.451	0.036	
Social	5000	1047	39	1.283	0.033	
Society	5000	636	27	1.692	0.063	

Experimental setup

1. Split 10% or 20 % data for training from data set.
2. Select parameters by performing 5-fold cross-validation on the training set.
3. Every experiment is repeated 20 times by randomly re-splitting the dataset into the training and the testing sets.

Evaluation Metrics

we choose F_1 measure as the evaluation metrics, which can be seen as the weighted average of F1 scores over all the categories.

$$F_1(s) = \frac{2p_s r_s}{p_s + r_s}$$

where,

ith instance x_i's predicted labels

precision →

$$\mathrm{p}_s = \frac{|\{x_i | s \in C_i \wedge s \in \hat{C}_i\}|}{|\{x_i | s \in \hat{C}_i\}|}$$

recall →

$$\mathrm{r}_s = \frac{|\{x_i | s \in C_i \wedge s \in \hat{C}_i\}|}{|\{x_i | s \in \mathrm{C}_i\}|}$$

ith instance x_i's true labels

Comparing methods

1. Binary Relevance method
2. Supervised Multi-label learning algorithm(e.g. CCE)
3. Constrained Non-negative Matrix Factorization(CNMF)
4. Dynamic Label Propagation for Semi-supervised Multi-class Multi-label Classification(DLP)
5. Semi-supervised Multi-label learning method by solving Sylvester Equation(SMSE)

Constrained Non-negative Matrix Factorization

The key assumption behind this work is that two examples tend to be assigned similar sets of class labels if they share high similarity in the input patterns.

Concretely, they expect $A_{i,j} \approx \mathbf{t}_i^T B \mathbf{t}_j$.

1. $A_{i,j}$ is the similarity of the ith and jth instance based on feature space.
2. $B_{k,l}$ is the similarity of the kth and lth category computed by labeled data.

CNMF

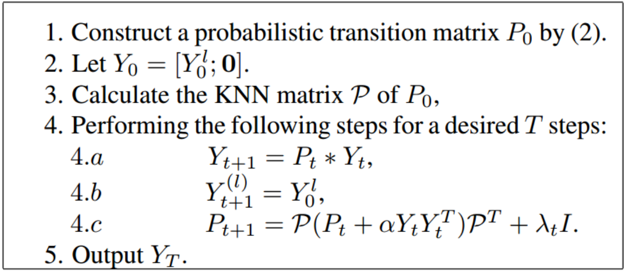
$$\begin{aligned} \arg\min_{\mathbf{T}} \quad & \sum_{i,j=1}^{n} \left(A_{i,j} - \sum_{k,l=1}^{m} T_{i,k} B_{k,l} T_{j,l} \right)^2 && (1) \\ \text{s. t.} \quad & T_{j,l} \geq 0,\ j = 1, \ldots, n,\ l = 1, \ldots, m \\ & T_{i,k} = \bar{T}_{i,k},\ i = 1, \ldots, n_l,\ k = 1, \ldots, m && (2) \end{aligned}$$

1. $\bar{T}_{i,\cdot}$ is the label vector of the ith labeled instance.

Dynamic Label Propagation for Semi-supervised Multi-class Multi-label Classification

1. Improved transition matrix by fusing information of both data features and data labels in each iteration.
2. Two instances with high correlated label vectors tend to have high similarity in the input data space.

DLP Algorithm



```
1. Construct a probabilistic transition matrix $P_0$ by (2).
2. Let $Y_0 = [Y_0^l; \mathbf{0}]$.
3. Calculate the KNN matrix $\mathcal{P}$ of $P_0$,
4. Performing the following steps for a desired $T$ steps:
   4.$a$    $Y_{t+1} = P_t * Y_t$,
   4.$b$    $Y_{t+1}^{(l)} = Y_0^l$,
   4.$c$    $P_{t+1} = \mathcal{P}(P_t + \alpha Y_t Y_t^T)\mathcal{P}^T + \lambda_t I$.
5. Output $Y_T$.
```

Figure 2. Algorithm of Dynamic Label Propagation (DLP).

Semi-supervised Multi-label learning method by solving Sylvester Equation

Two graphs are first constructed on instance level and category level respectively.

1. For instance level, each node represents one instance and each edge weight reflects the similarity between corresponding pairwise instances.
2. For category level, each node represents one category and each edge weight reflects the similarity between corresponding pairwise categories.

Sylvester equation

$$\underset{\text{m x m}}{\mathrm{A}}\underset{\text{m x n}}{\mathrm{X}} + X\overset{\text{n x n}}{\mathrm{B}} = \underset{\text{m x n}}{\mathrm{C}}$$

1. A Sylvester equation has a unique solution for X exactly when there are no common eigenvalues of A and -B.
2. A classical algorithm for the numerical solution of the Sylvester equation is the Bartels Stewart algorithm.

SMSE

instance level energy function

$$\min \infty \sum_{i=1}^{l} ||f_i - y_i||^2 + \mu \, \mathrm{E(f)} + \nu \mathrm{E'(g)}$$

category level energy function

where,

$$E(f) = \tfrac{1}{2} \sum_{i,j=1}^{n} W_{ij} ||f_i - f_j||^2$$

$$E'(g) = \tfrac{1}{2} \sum_{i,j=1}^{k} W'_{ij} ||g_i - g_j||^2$$

$(f_1, \ldots, f_n)^T = (g_1, \ldots, g_k)$

The end
Thank you!