Online Transfer with Heterogeneous Source

Yan Yuguang

South China University of Technology

May 11, 2015

The goal is to learn some prediction function $f(\mathbf{x}_t)$ on a target domain in an online fashion from a sequence of instances $\{(\mathbf{x}_t, y_t | t = 1, 2, \cdots, T\}$ in data space $\mathcal{X} \times \mathcal{Y}$.

- Homogeneous source domain:

$$\mathcal{X} = \mathcal{X}^k, \mathcal{Y} = \mathcal{Y}^k$$

- Heterogeneous source domain:

$$\mathcal{X} \cap \mathcal{X}^k = \emptyset, \mathcal{Y} = \mathcal{Y}^k$$

co-occurrence information (?, ?)

- Build text features for image [1]
- annotation-based probabilistic latent semantic analysis [2]
- Construct representation for image using common semantic view between image and text data [3]
- Co-transfer via transition probability [4] [5]

Above-mentioned studies require prior training data in the source and target domains.
In our problem setting, all the instance in the target domain arrive sequentially.

[1] Building text features for object image classification (?, ?)
[2] Heterogeneous Transfer Learning for Image Clustering via the Social Web (?, ?)
[3] Heterogeneous Transfer Learning for Image Classification (?, ?)
[4] Co-Transfer Learning via Joint Transition Probability Graph Based Method (?, ?)
[5] Cotransfer Learning Using Coupled Markov Chains with Restart (?, ?)

Only a few works consider transfer learning in an online fashion.

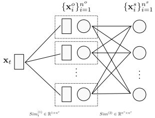
- Ensemble learning method for online homogeneous transfer [6] [7]
- Multi-view method for online heterogeneous transfer

Zhao et al. assume that the feature space of the source domain is a subset of that of the target domain.
In our problem setting, the feature spaces of the source and target domains do not share any common subset (e.g. image and text).

[6]OTL: A Framework of Online Transfer Learning (?, ?)
[7]Online Transfer Learning (?, ?)

Problem Definition

test



Similarity between $\mathbf{x}_t$ and each instance in the heterogeneous source domain

$$Sim_t(j) = \sum_i Sim_t^{(1)}(i) Sim^{(2)}(i,j)$$

Weighted K nearest neighbor classifier

$$h_t^s(\mathbf{x}_t) = \sum_{i\in N} \frac{Sim_t(i)}{\sum_{i\in N} Sim_t(i)} y_i$$

where N is the set of identifiers of K nearest neighbors in the heterogeneous source domain.

Algorithm 1 Online Heterogeneous Transfer Algorithm 1 (OHT1)

Input: aggressiveness parameter $C > 0$
$\eta = \frac{1}{2}$
heterogeneous source data
Output: $\mathbf{v} = \mathbf{0}$, $w_1^s \in (0,1)$, $w_1 \in (0,1)$, where $w_1^s + w_1 = 1$

1: **for** $t = 1$ to T **do**
2: receive instance: $\mathbf{x}_t \in \mathcal{X}$
3: normalize: $\theta_t^s = \frac{w_t^s}{w_t^s + w_t}, \theta_t = \frac{w_t}{w_t^s + w_t}$
4: predict: $\hat{y}_t = \text{sign}\left(\theta_t^s \Omega(h_t^s(\mathbf{x}_t)) + \theta_t \Omega(\mathbf{v}_t \cdot \mathbf{x}_t) - \frac{1}{2}\right)$
5: receive correct label: $y_t \in \mathcal{Y}$
6: compute:

Algorithm 2 Online Heterogeneous Transfer Algorithm 2 (OHT2)

Input: aggressiveness parameter $C > 0$
discount parameter $\alpha \in (0,1)$
heterogeneous source data
Output: $\mathbf{v} = \mathbf{0}$, $w_1^s \in (0,1)$, $w_1 \in (0,1)$, where $w_1^s + w_1 = 1$

1: **for** $t = 1$ to T **do**
2: receive instance: $\mathbf{x}_t \in \mathcal{X}$
3: normalize: $\theta_t^s = \frac{w_t^s}{w_t^s + w_t}, \theta_t = \frac{w_t}{w_t^s + w_t}$
4: predict: $\hat{y}_t = \text{sign}\left(\theta_t^s \,\text{sign}(h_t^s(\mathbf{x}_t)) + \theta_t \,\text{sign}(\mathbf{v}_t \cdot \mathbf{x}_t)\right)$
5: receive correct label: $y_t \in \mathcal{Y}$
6: compute:

$$w_{t+1}^s = w_{t+1}^s \alpha^{I(y_t h_t^s(\mathbf{x}_t) \leq 0)}$$

At t-th trial, $\mathbf{Hedge}(\beta)$ algorithm synthesizes opinions from different experts based on a weight vector $\mathbf{weight}_t$, and updates the weight vector using rule

$$weight_{t+1}^{i} = weight_{t}^{i} \cdot \beta^{loss_t^i}$$

where $\beta \in [0,1]$ and $loss_t^i \in [0,1]$.

$\mathbf{Hedge}(\beta)$	OHT1	OHT2
$loss_t^i$	$\ell_t^s = (\Omega(h_t^s(\mathbf{x}_t)) - \Omega(y_t))^2$ $\ell_t = (\Omega(\mathbf{v}_t \cdot \mathbf{x}_t) - \Omega(y_t))^2$	$\ell_t^s = I(y_t h_t^s(\mathbf{x}_t) \leq 0)$ $\ell_t = I(y_t(\mathbf{v}_t \cdot \mathbf{x}_t) \leq 0)$
β	$\beta = \exp\{-\eta\}$	$\beta = \alpha$

OHT algorithms obey the rule of $\mathbf{Hedge}(\beta)$ algorithm.

Proposition

Given loss $\ell_t^s \in [0,1]$, $\ell_t \in [0,1]$ *and decay factor* $\beta \in (0,1)$, *for any sequence of loss vectors* $\{(\ell_t^s, \ell_t) | t = 1, 2, \cdots T\}$, *we have*

$$\sum_{t=1}^{T} (\theta_t^s \ell_t^s + \theta_t \ell_t) \leq \frac{1}{1-\beta} \min\left(\Delta^s, \Delta\right)$$

where $\Delta^s = \ln(\frac{1}{w_1^s}) + (\ln \frac{1}{\beta}) \sum\limits_{t=1}^{T} \ell_t^s$ *and*

$\Delta = \ln(\frac{1}{w_1}) + (\ln \frac{1}{\beta}) \sum\limits_{t=1}^{T} \ell_t$

Theorem

Let M be the number of mistakes made by OHT1 algorithm, then we have

$$M \leq \frac{4}{1-\beta}\min(\Delta^s, \Delta)$$

where $\Delta^s = \ln(\frac{1}{w_1^s}) + (\ln\frac{1}{\beta})\sum_{t=1}^{T}\ell_t^s$ *and*

$\Delta = \ln(\frac{1}{w_1}) + (\ln\frac{1}{\beta})\sum_{t=1}^{T}\ell_t$.

Theorem

Let M be the number of mistakes made by OHT2 algorithm, then we have

$$M \leq \frac{2}{1-\beta}\min(\Delta^s, \Delta)$$

NUS-WIDE dataset

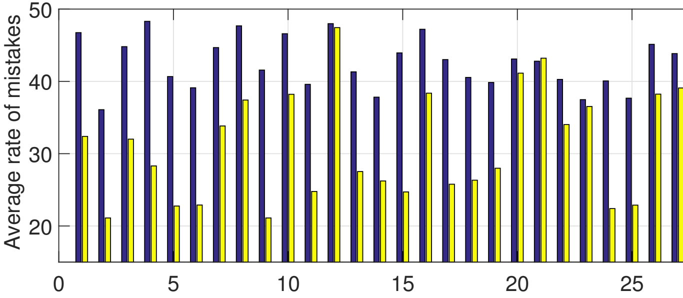
- Target domain: Image
- Heterogeneous source: Text
- Co-occurrence data: co-occurred image-tag pairs

Experiments

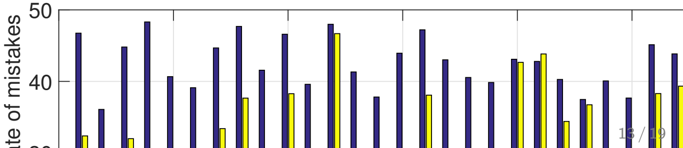
Baseline Methods

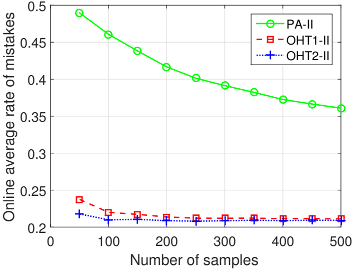
- Passive-Aggressive algorithms
  Do not exploit knowledge from the source domain
- Kernel function
  Gaussian Kernel
- Number of nearest neighbors
  $K = 100$

Experiments
Results

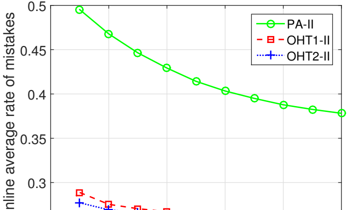


(a) PA-II vs. OHT1-II




Online average rate of mistakes
0.5
0.45
0.4
0.35
0.3
0.25
0.2
0
100
200
300
400
500
Number of samples
PA-II
OHT1-II
OHT2-II

(a) Task 2

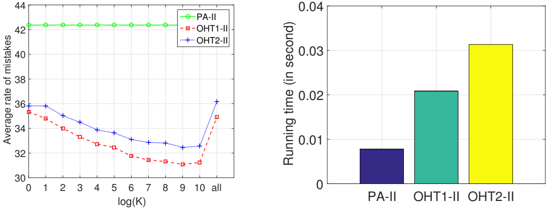

nline average rate of mistakes
0.5
0.45
0.4
0.35
0.3
PA-II
OHT1-II
OHT2-II

Paired t-test ($\alpha = 0.01$)

- OHT1 vs. PA: 44/0/1
- OHT2 vs. PA: 42/2/1

Cohen's d value ($d > 0.8$: large promotion, $0.2 < d < 0.8$: middle promotion)

- OHT1: 41/3
- OHT2: 40/3



(a) (b)

Figure : (a) The average rate of mistakes under varying values of K. (b) The average running time of different algorithms when all instances in heterogeneous source are considered.

Conclusion and Future Works

- We explore online heterogeneous transfer learning problem.
- We construct a connection across the domains using co-occurrence data, and apply the ensemble strategy to train a classifier.
- We offer the theoretical analysis of our algorithms.
- Experimental results show the effectiveness of our algorithms.

Future works:

- applications with other types of data
- multiple source domains

THANK YOU FOR YOUR ATTENTION!