Online Transfer Learning with Heterogeneous Source

Author1
Address line

Author2
Address line

Author3
Address line

Abstract

AAAI creates proceedings, working notes, and technical reports directly from electronic source furnished by the authors. To ensure that all papers in the publication have a uniform appearance, authors must adhere to the following instructions.

Introduction

Related Work

Transfer Learning

Online Learning

Problem Definition

Suppose that we are given some text instance $\{\mathbf{x}_i^s, y_i^s\}$ in the source data space $\mathcal{X}^s \times \mathcal{Y}^s$, where $\mathcal{X}^s = \mathbb{R}^{d^s}$ and $\mathcal{Y}^s = \{+1, -1\}$. The objective of online transfer learning is to learn a classifier $f(\mathbf{x}_t)$ to classify image instance on a target domain in a online fasion. The data space of the target domain is $\mathcal{X} \times \mathcal{Y}$, where $\mathcal{X} = \mathbb{R}^d$ and $\mathcal{Y} = \{+1, -1\}$. Specifically, the task of online heterogeneous transfer learning is a sequential process, during which an image instance $\mathbf{x}_t$ comes at the t-th trial, and the classifier generates a predicted class label $\hat{y}_t$. Then the classifier receives the true class label y_t and update itself to obtain better classification ability.

In our problem setting, $\mathcal{Y} = \mathcal{Y}^s$, which means the same class label in two domains indicates the same class, for instance, $+1$ indicates vehicle and -1 indicates tree. However, the motivation of us is to conduct online image classification task by leveraging knowledge in text data. Formally, $\mathcal{X} \cap \mathcal{X}^s = \varnothing$. We cannot directly transfer information from a completely different source domain. Therefore, a sophistecated knowledge transfer method is proposed.

Methods

In this section, we describe the details of our proposed algorithms. We extend HomOTL algorithms, which exploit knowledge in a homogeneous source domain, to tackle the online heterogeneous transfer learning problem. We first introduce our approach of how to establish relationship between image data and auxiliary text data to build classifier using heterogeneous source data. Then we present online heterogeneous transfer learning algorithms. Finally, we analyze the theoretical bounds of the algorithms in a unified framework based on Hedge(β) algorithm.

Heterogeneous Knowledge Transfer

Algorithms

Given a hypothesis obtained from the hetegeneous source domain, we can utilize the ensemble learning stragety similar to HomOTL algorithms.

We first construct a prediction function in the target domain using online learning algorithm PA, and then combine two hypotheses to get final classifier. According to the loss value suffered by each hypothesis seperately, we dynamically update two weights of both hypotheses. Without loss of generality, we use linear function in the target domain to simplify the description. Through the kernel trick, we are able to tackle the non-linear classification problem.

Algorithm 1 presents the process of the proposed online heterogeneous transfer learning algorithm 1 (OHT1). θ_1^s and θ_1 reflect the prior confidence we have on two hypotheses, and must sum to 1. If we have no preference on any of them, we can simply set $\theta_1^s = \theta^t = \frac{1}{2}$, which is also the setting we used in our experiments. Function $\Omega(z) = \max\{0, \min\{1, \frac{z+1}{2}\}\}$ projects $z \in \mathbb{R}$ to range $[0, 1]$. In order to adjust two weights dynamically, exponentially weighting update method cite is used.

It is worthwhile to note that the equation in step 8, which calculate the updating factor of linear classifier trained on the target domain, can be replaced by two variants. In cite, PA-I introduces a non-negative linear slack variable into PA, and PA-II introduces a quadratic slack variable. The corresponding equations of calculation of τ are $\tau_t = \min\{C, \frac{\ell_t}{\|\mathbf{x}_t\|^2}\}$ and $\tau_t = \frac{\ell_t}{\|\mathbf{x}_t\|^2 + \frac{1}{2C}}$ respectively. Likewise, by replacing the equation in Step 8, we can obtain two variants OHT1-I and OHT1-II.

Algorithm 2 summarizes the online heterogeneous transfer learning 2 (OHT2). OHT2 has a more polarized attitude to two hypotheses. Indication function $I(\cdot)$ represents whether the hypothesis makes a wrong prediction or not.

Algorithm 1 Online Heterogeneous Transfer Algorithm 1 (OHT1)

Input: aggressiveness parameter $C > 0$
heterogeneous source data
Initialize: $\mathbf{v} = \mathbf{0}, \theta_1^s \in (0,1), \theta_1 \in (0,1)$

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

$$\theta_{t+1}^s = \theta_{t+1}^s \exp\left\{-\eta\big(\Omega(h_t^s(\mathbf{x}_t)) - \Omega(y_t)\big)^2\right\}$$

$$\theta_{t+1} = \theta_{t+1} \exp\left\{-\eta\big(\Omega(\mathbf{v}_t \cdot \mathbf{x}_t) - \Omega(y_t)\big)^2\right\}$$

  suffer loss: $\ell_t = \max\{0, 1 - y_t(\mathbf{v}_t \cdot \mathbf{x}_t)\}$
  set: $\tau_t = \frac{\ell_t}{\|\mathbf{x}_t\|^2}$
  update: $\mathbf{v}_{t+1} = \mathbf{v}_t + \tau_t y_t \mathbf{x}_t$
end for

The mistake made by a hypothesis will decrease its weight in the next trial.

Similar to OHT1 algorithms, we also have two variants OHT2-I and OHT2-II.

Algorithm 2 Online Heterogeneous Transfer Algorithm 2 (OHT2)

Input: aggressiveness parameter $C > 0$
discount parameter $\alpha \in (0,1)$
heterogeneous source data
Initialize: $\mathbf{v} = \mathbf{0}, \theta_1^s \in (0,1), \theta_1 \in (0,1)$

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

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

$$\theta_{t+1} = \theta_{t+1} \alpha^{I(y_t(\mathbf{v}_t \cdot \mathbf{x}_t) \le 0)}$$

  suffer loss: $\ell_t = \max\{0, 1 - y_t(\mathbf{v}_t \cdot \mathbf{x}_t)\}$
  set: $\tau_t = \frac{\ell_t}{\|\mathbf{x}_t\|^2}$
  update: $\mathbf{v}_{t+1} = \mathbf{v}_t + \tau_t y_t \mathbf{x}_t$
end for

Theoretical Analysis

Experimental Results

In this section, we empirically evaluate the performance of proposed online heterogeneous transfer learning algorithms and classic online Passive-Aggressive algorithms, which consists of a original version PA and its two variations PA-I and PA-II. Encouraging results demonstrate that the proposed algorithms outperform baseline methods.

Dataset

Our experiments are conducted for image classification by leveraging information from text data. We use NUS-WIDE dataset to generate learning tasks. The NUS-WIDE dataset is extracted from Flickr. It includes 269,648 images and the associated tags from Flickr, with a total number of 5,018 unique tags. An image instance is represented by a feature vector based on SIFT descriptions, and a text instance is represented by a feature vector based on tags. There are 81 ground-truth class labels in the dataset. We randomly selected 10 classes (bird, boat, car, flower, food, rock, sun, toy, tree) and built $C_{10}^2 = 45$ binary classification tasks.

We refer the images as data in the target domain, and the tags as the text data in the heterogeneous source domain. Each binary classification task has 500 image instances in the target domain, 1,200 text instances in the heterogeneous source domain, and 1,500 co-occurred image-text pairs. In order to obtain stable results, we draw 100 times of random permutation of the image instances in the target domain and evaluate the performance of learning algorithms based on average rate of mistakes.

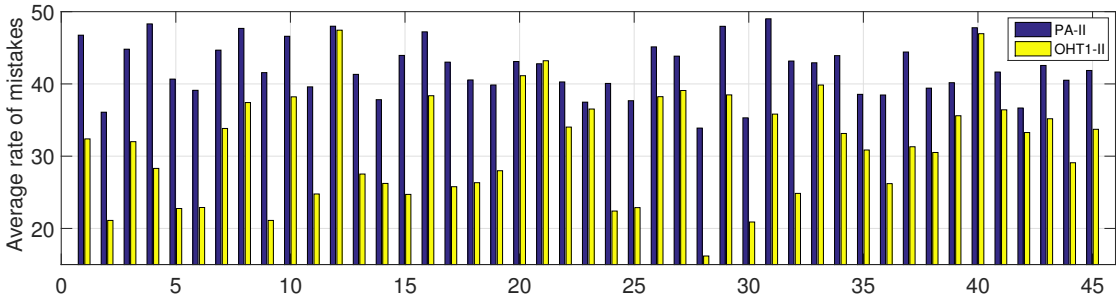
Baseline Methods

We compare the proposed methods with Passive-Aggressive online learning algorithms. PA algorithm proposed by Crammer et al. does not exploit knowledge from the source domain. It deals with the traditional online learning problem in the target domain. In addition, PA-I introduces a non-negative linear slack variable into PA, and PA-II introduces a quadratic slack variable. Likewise, we have three versions of tow OHT algorithms (OHTi, OHTi-I and OHTi-II, where $i = \{1, 2\}$) respectively based on PA algorithm and its variations. We conduct three sets of experiments considering three versions separately. Each set of experiments compares two OHT methods against a Passive-Aggressive algorithm.

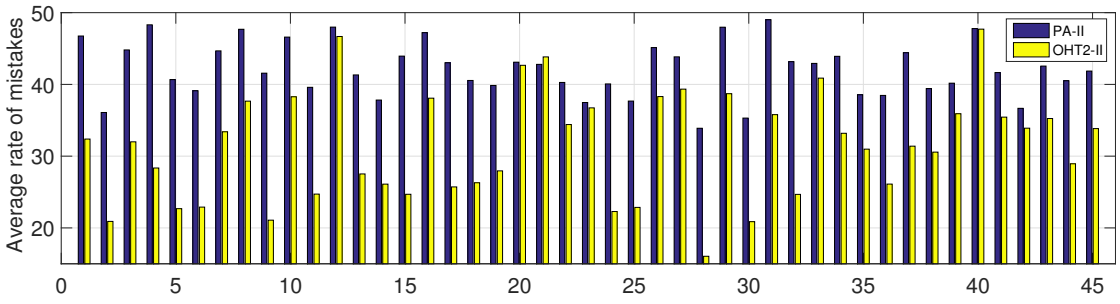
For fair comparison and simplicity, we adopt Gaussian kernel function on all the algorithms and tasks. The kernel parameter $\sigma = 8$ for the target domain. The regularization parameter $C = 5$, $\beta = \frac{\sqrt{T}}{\sqrt{T} + \sqrt{2 \ln 4}}$ for OHT2 algorithm. In addition, we set the number of nearest neighbors to be considered $K = 100$. Sensitivity of parameters will be examined in subsequent sections.

Results and Discussion

Figure 1 summarizes the mistake rates of all 45 binary classification tasks in the third set, which compares OHT1-II, OHT2-II and PA-II. The x-axis of the figure refers to the 45 tasks. We see that on most tasks, PA-II has the very high mistake rate, which prove the dificulty of image classification task without any auxiliary source information and the necessity of knowledge transfer. The observation that our proposed OHT methods in general outperform Passive-Aggressive algorithm validates the effectivity of heterogeneous transfer learning. Similar experimental results are observed in other two sets. Because of the restricted space, we are not able to report them. In order to facilitat the description, we denote PA-II, OHT1-II and OHT2-II by PA, OHT1 and OHT2 respectively in the following discussion.

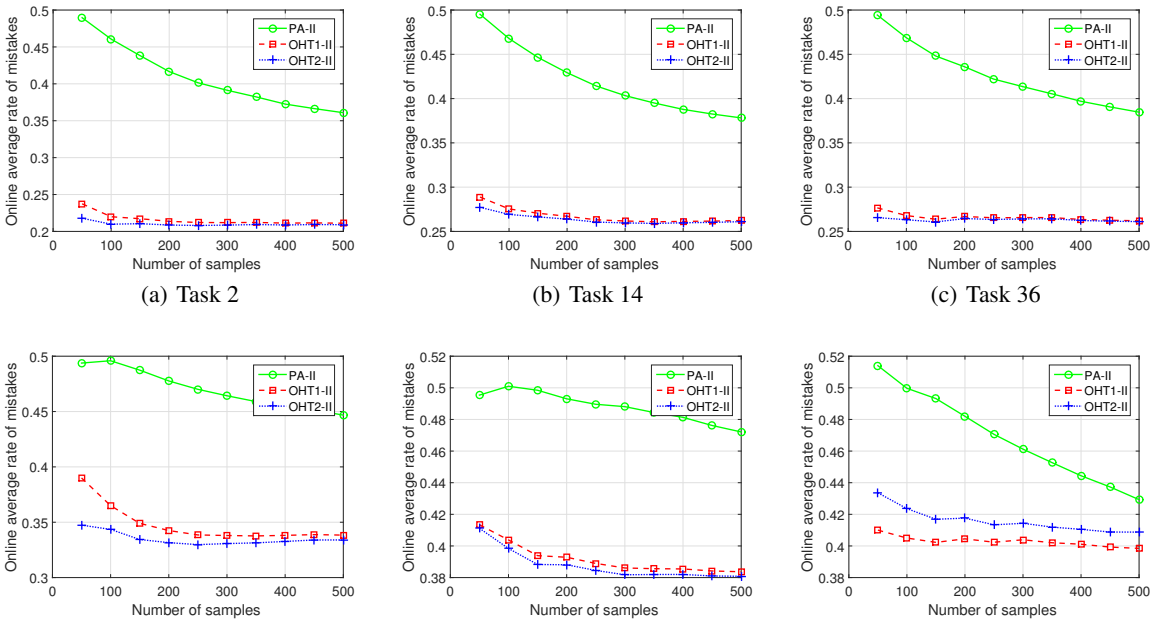


(a) PA-II vs. OHT1-II



(b) PA-II vs. OHT2-II

Figure 1: Average rate of mistakes on all 45 tasks



(a) Task 2 (b) Task 14 (c) Task 36

(d) Task 7 (e) Task 16 (f) Task 33

Figure 2: Online average rate of mistakes on example tasks

Figure 2 illustrates the dynamic process of several representative online learning tasks, respectively. We observe that OHT algorithms usually achieve better performance at the beginning stage. On some tasks(e.g., 7, 16 and 33), the online mistake rates of all three algorithms decrease during the period, and OHT methods always obtain better performance than PA algorithm. These observations verifies that the OHT algorithms indeed transfer useful knowledge from the heterogeneous source domain to the target domain.

We also analyze the performance difference between PA

and two OHT algorithms. Statistical significance against PA was assessed by paired t-test at 0.01 level. For each task, a win (or loss) is counted when OHT algorithm is significatly better (or worse) than PA algorithm over 100 trials. Otherwise, a tie is recorded. The win/tie/loss results is 44/0/1 for competition between OHT1 and PA, and 42/2/1 for competition between OHT2 and PA. This result validates that our OHT algorithms is statistically better than PA algorithm.

Besides, we utilize Cohen's d value to measure the improvement of our algorithms. Generally, $d > 0.8$ indicates a large promotion, and $0.2 < d < 0.8$ indicates a middle promotion. In our experiments, OHT1 algorithm achieves large improvement on 41 tasks and middle improvement on 3 tasks. For OHT2 algorithms, the numbers are 40 and 3. Combining the win/tie/loss results, we see that OHT1 is more stable than OHT2.

Parameters and Running time

Parameters Experiments in paper about online transfer learning illustrated that the performance of online transfer learning algorithms is generally insensitive to the parameter C and β. Consequently, we only investigate how different values of parameter K affect the mistake rates of the algorithms. Figure 3(a) shows the average mistake rates with varied values of parameter K over all 45 tasks. PA algorithm, whose performance is not related to the parameter K, provide a baseline rate of mistakes. We observe that the performance of the proposed methods consistently outperform PA algorithm, which indicates that nearest neighbors in heterogeneous source domain do provide valuable advice for the classification task.

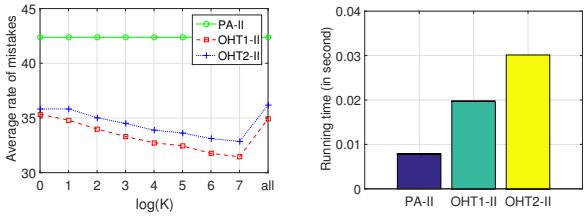


Figure 3: (a) The average rate of mistakes under varying values of K. (b) The average running time of different algorithms when all instances in heterogeneous source are considered.

Running time All of the algorithms were implemented in Matlab, and all experiments were run in a Linux machine with 3.2 GHz CPU and 3.8 GB memory. Compared to PA algorithm without exploiting any information from the source domain, OHT algorithms are less efficient. The main reason of more running time for OHT algorithms is probably the searching process for the nearest neighbors. We can simply make use of all instances in the heterogeneous source domain to get rid of overhead for searching nearest neighbors. Figure 3(b) shows the running time of different algorithms when all instances in the heterogeneous source domain are considered. We obtain generally comparable running time to PA, and at the same time, achieve better performance than PA.

Conclusion