Online Heterogeneous Transfer Learning by Weighted Offline and Online Classifiers

Yuguang Yan, Qingyao Wu(✉), Mingkui Tan(✉), and Huaqing Min

School of Software Engineering, South China University of Technology, Guangzhou, China
qyw@scut.edu.cn, mingkuitan@scut.edu.cn

Abstract. In this paper, we study online heterogeneous transfer learning (HTL) problems where offline labeled data from a source domain is transferred to enhance the online classification performance in a target domain. The main idea of our proposed algorithm is to build an offline classifier based on heterogeneous similarity constructed by using labeled data from a source domain and unlabeled co-occurrence data which can be easily collected from web pages and social networks. We also construct an online classifier based on data from a target domain, and combine the offline and online classifiers by using the Hedge weighting strategy to update their weights for ensemble prediction. The theoretical analysis of error bound of the proposed algorithm is provided. Experiments on a real-world data set demonstrate the effectiveness of the proposed algorithm.

1 Introduction

Heterogeneous Transfer Learning (HTL) aims to transfer knowledge from a source domain with sufficient labeled data to enhance learning performance on a target domain where the source and target data are from different feature spaces [10,12,17,18]. It has been shown that the learning performance of HTL tasks can be significantly enhanced if co-occurrence data is considered [5,9,12,13,15–18,21]. Co-occurrence data is cheap and easily collected from web pages or social networks. For example, the target learning task is image classification, a set of labeled text documents is given as auxiliary data in a source domain, and we can easily collect some text and image co-occurrence data (such as image annotations or documents around images) for text-to-image heterogeneous transfer learning.

Most existing studies of HTL work on offline/batch learning fashion, in which all the training instances from a target domain are assumed to be given in advance. However, this assumption may not be valid in practice where target instances are received one by one in an online/sequential manner. Unlike the previous studies, we investigate HTL under an online setting [1,8]. For instance, we consider an image classification task for user generated content on some social computing applications. The social network users usually post pictures and attach some text comments for the pictures. The text data is given as a source

G. Hua and H. Jégou (Eds.): ECCV 2016 Workshops, Part III, LNCS 9915, pp. 467–474, 2016.
DOI: 10.1007/978-3-319-49409-8_38

domain data and text-image pairs are considered as co-occurrence information, and the task is to classify new image instances sequentially in a target domain. The crucial issue is how to effectively use offline text data and text-image pairs to improve the online image classification performance.

There are only a few research works that address online transfer learning problems. In [7,14,19,20], researchers studied online homogenerous transfer learning problems where source and target instances are represented in the same feature space. For online heterogeneous setting, existing methods are based on the assumption that the feature space of the source domain is a subset of that of the target domain [19,20].

Motivated by recent research in transfer learning and online learning, in this paper, we study online heterogeneous transfer learning (HTL) problems where labeled data from a source domain and unlabeled co-occurrence data from auxiliary information are under offline mode and data from a target domain is under online mode. We propose a novel method called *Online Heterogeneous Transfer with Weighted Classifiers* (OHTWC) to deal with this learning problem (see Fig. 1). In OHTWC, we build an offline classifier based on heterogeneous similarity constructed by using labeled data from a source domain and unlabeled co-occurrence data from auxiliary information, and construct an online classifier based on data from a target domain. The offline and online classifiers are then combined by using the **Hedge** (β) method [6] to make ensemble prediction dynamically. The theoretical analysis of the error bound of the proposed method is also provided.

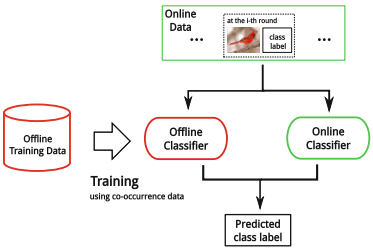


Fig. 1. Overall heterogeneous transfer learning system using offline and online classifiers.

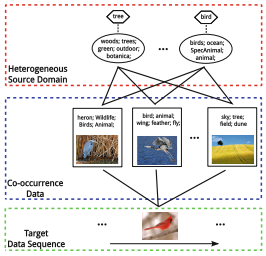


Fig. 2. Heterogeneous knowledge transfer based on co-occurrence data.

2 The Proposed Method

We study online heterogeneous transfer learning (HTL) problems where labeled instances $\{(\mathbf{x}_i^s, y_i^s)\}_{i=1}^{n^s} \in \mathcal{X}^s \times \mathcal{Y}^s$ from a source domain and unlabeled co-occurred pairs $\{(\mathbf{u}_i^c, \mathbf{v}_i^c)\}_{i=1}^{n^c} \in \mathcal{X}^c$ from an auxiliary information are under offline mode and instances $\{(\mathbf{x}_i, y_i)\}_{i=1}^{n} \in \mathcal{X} \times \mathcal{Y}$ from a target domain is under online mode. Here n^s and n^c refer to the number of labeled instances

in the source domain and the number of co-occurred pairs. The feature space $\mathcal{X}^s$ of the source domain is different from the feature space $\mathcal{X}$ of the target domain. The class labels are the same as in both source and target domains, i.e., $\mathcal{Y}^s = \mathcal{Y} = \{+1, -1\}$. There are two components $\mathbf{u}_i^c$ and $\mathbf{v}_i^c$ in the co-occurred pair where $\mathbf{u}_i^c$ belongs to $\mathcal{X}^s$ and $\mathbf{v}_i^c$ belongs to $\mathcal{X}$. The objective of online HTL is to learn an online classifier $f(\mathbf{x}_i)$ to generate a predicted class label $\hat{y}_i$ where the instance $\mathbf{x}_i$ arrives at the i-th trial. The classifier then receives the correct class label y_i and update itself according to their difference to obtain a better classification ability.

2.1 The Offline Classifier

The offline classifier is based on the similarity relationship between the instances in the source and target domains via the co-occurrence data. The idea of similarity calculation is given in a text-image classification example in Fig. 2. In this example, we have target image instances which arrive in an online manner, labeled text data in the heterogeneous source domain under an offline setting, and unlabeled co-occurred pairs which provide information between text data in the source domain and image data in the target domain.

When the j-th instance $\mathbf{x}_j$ arrives in the target domain, we make use of the Pearson correlation to measure the similarity $a_j(i)$ between $\mathbf{x}_j$ and $\mathbf{v}_i^c$: $a_j(i) = \frac{(\mathbf{x}_j - \bar{\mathbf{x}}_j)^\top (\mathbf{v}_i^c - \bar{\mathbf{v}}_i^c)}{\|\mathbf{x}_j - \bar{\mathbf{x}}_j\| \|\mathbf{v}_i^c - \bar{\mathbf{v}}_i^c\|}, 1 \le i \le n^c$, where $\bar{\mathbf{z}}$ is a vector whose all elements are equal to mean($\mathbf{z}$) (i.e., the mean value of all elements of vector $\mathbf{z}$), and $\|\cdot\|$ is the Euclidean distance. Similarly, we construct the similarity $b_l(i)$ between $\mathbf{x}_l^s$ and $\mathbf{u}_i^c$: $b_l(i) = \frac{(\mathbf{x}_l^s - \bar{\mathbf{x}}_l^s)^\top (\mathbf{u}_i^c - \bar{\mathbf{u}}_i^c)}{\|\mathbf{x}_l^s - \bar{\mathbf{x}}_l^s\| \|\mathbf{u}_i^c - \bar{\mathbf{u}}_i^c\|}, 1 \le i \le n^c$, Therefore, we compute the similarity $r_j(l)$ between $\mathbf{x}_j$ and $\mathbf{x}_l^s$ via co-occurred pairs as follows: $r_j(l) = \sum_{i=1}^{n^c} a_j(i) b_l(i), 1 \le l \le n^s$. According to $r_j(l)$, we can make a prediction $h^s(\mathbf{x}_j)$ for the given instance $\mathbf{x}_j$ by computing the weighted sum of the labels of its k nearest neighbors from the source domain:

$$h^s(\mathbf{x}_j) = \Big(\sum_{k \in N} r_j(k) y_k^s\Big) \Big/ \Big(\sum_{k \in N} r_j(k)\Big), \tag{1}$$

where the set N includes indices of $\mathbf{x}_j$'s k nearest neighbors that are found in the source domain.

2.2 The Online Classifier

Besides the classifier $h^s(\mathbf{x}_i)$ obtained from the heterogeneous source domain, we construct another classifier $h_i(\mathbf{x}_i) = \mathbf{w}_i^\top \mathbf{x}_i$ by using target instances based on online learning algorithm (PA) [4,11]. The PA algorithm models online learning as a constrained convex optimization problem, and updates the classifier as follows

$$\mathbf{w}_{i+1} = \mathbf{w}_i + \tau_i y_i \mathbf{x}_i, \tag{2}$$

where $\tau_i = \min\left\{c, \frac{\ell^*(\mathbf{x}_i, y_i; \mathbf{w}_i)}{||\mathbf{x}_i||^2}\right\}$, c is a positive regularization parameter, and $\ell^*(\mathbf{x}, y; \mathbf{w}) = \max\{1 - y(\mathbf{w}^\top \mathbf{x}), 0\}$ is the hinge loss.

2.3 Hedge(β) Strategy for Weighted Classifers

We propose to combine the offline and online classifiers suitably such that the resulting classification performance can be enhanced. In this paper, we make use of the **Hedge**(β) strategy [6] to update the weights of offline and online classifiers dynamically. Let ℓ_i^s and ℓ_i be the loss values that are generated by $h^s(\mathbf{x}_i)$ and $h_i(\mathbf{x}_i)$, respectively. The **Hedge**(β) stragtey is used to generate the positive weights θ_i^s and θ_i for $h^s(\mathbf{x}_i)$ and $h_i(\mathbf{x}_i)$ such that the resulting prediction is given by

$$\hat{y}_i = \text{sign}\left(\theta_i^s \phi(h^s(\mathbf{x}_i)) + \theta_i \phi(h_i(\mathbf{x}_i)) - \frac{1}{2}\right), \tag{3}$$

where $\theta_i^s + \theta_i = 1$, and ϕ is a predefined function that maps the predicted value into range $[0, 1]$ The two weights (i.e., θ_i^s and θ_i) are updated by using the following rules:

$$\theta_{i+1}^s = \theta_i^s \beta^{\psi(y_i h^s(\mathbf{x}_i))}, \quad \theta_{i+1} = \theta_i \beta^{\psi(y_i h_i(\mathbf{x}_i))}, \tag{4}$$

where $\beta \in (0, 1)$ and ψ is also a predefined loss function for controlling the update of the weights. We see in (4) that a larger loss will result in a larger decay, thus the better classifier will relatively obtain a larger weight value.

For simplicity, let h be the predicted value (i.e., $h_i(\mathbf{x}_i)$ or $h^s(\mathbf{x}_i)$), We design the following mapping function

$$\phi(h) = \frac{1}{1 + \exp\{-h\}}; \quad \psi(yh) = \frac{1}{1 + \exp\{yh\}}. \tag{5}$$

The loss value is dependent on the predicted result and the confidence we have on the predicted value. The absolute value $|h|$ measures the confidence we have on the predicted result. On the other hand, $\psi(yh)$ maps the margin value yh into range $[0, 1]$, leading the decay of the weights of classifiers. In general, when we get a margin with a large absolute value, if our prediction is correct, we will obtain a small loss. However, if our prediction is incorrect, we have to suffer a large loss because of our wrong guess.

2.4 Theoretical Analysis

Theorem 1. Define $\ell_i^s = \psi(y_i h^s(\mathbf{x}_i))$, $\ell_i = \psi(y_i h_i(\mathbf{x}_i))$, and $\beta \in (0, 1)$ is the decay factor. Given $\theta_1 = \theta_1^s = \frac{1}{2}$. Let M be the number of mistakes made by the OHTWC algorithm after receiving a sequence of T instances, then we have

$$M \leq \frac{2}{1 - \beta} \min\{\Delta^s, \Delta\}, \tag{6}$$

where $\Delta^s = \ln 2 + (\ln \frac{1}{\beta}) \sum_{i=1}^{T} \ell_i^s$ and $\Delta = \ln 2 + (\ln \frac{1}{\beta}) \sum_{i=1}^{T} \ell_i$.

Remark. Theorem 1 states that the entire number of mistakes, which sums up the error at all T trials, is not much larger than the loss value made by the better single classifier.

3 Experiments

3.1 Data Set and Baseline Methods

We use the NUS-WIDE data set [3] as text-to-image online heterogeneous transfer learning data set. We refer the images as the data in the target domain, and the text instances as the auxiliary data in the heterogeneous source domain. Images and their corresponding tag data are used as the co-occurrence data. We randomly select 10 classes to build $\binom{10}{2} = 45$ binary image classification tasks. For each binary classification task, we randomly pick up 600 image instances, 1,200 text instances, and 1,600 co-occurred image-text pairs.

We compare our proposed algorithms with the PA [4], SVM [2], HTLIC [21] and HET algorithms. PA is used as a baseline method without knowledge transfer. To fit the online setting, we periodically train the SVM classifier when $\frac{T}{20}$ new target instances arrive, and use the trained classifier to make predictions for the next $\frac{T}{20}$ coming instances, where T is the total number of the target data. And HTLIC is adjusted to online learning problems. Specifically, PA algorithm is conducted on new features constructed by the approach in HTLIC. HET finds the nearest neighbors of each target instance in the co-occurrence data, and uses the heterogeneous views of the neighbors as the new representation of the target instance; then PA algorithm is performed on these heterogeneous new features.

We set the regularization parameter $c = 1$ for all the algorithms, $\beta = \frac{\sqrt{T}}{\sqrt{T}+\sqrt{2\ln 2}}$ for OHTWC, and the number of the nearest neighbors to $k = \frac{n^c}{10}$, where n^c is the number of co-occurrence data. In order to obtain stable results, we draw 20 times of random permutation of the data set and evaluate the performance of learning algorithms based on average rate of mistakes.

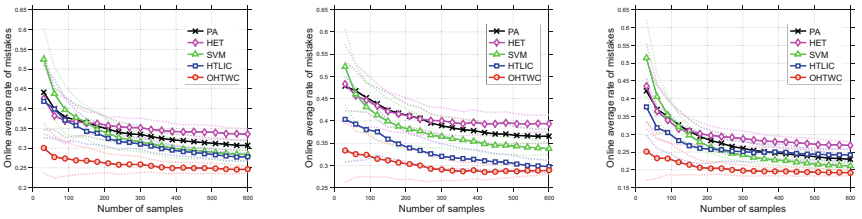
3.2 Results and Discussion

In Table 1, we present numerical results of all adopted algorithms on several representative tasks and the average results over all 45 tasks. We see that on average, SVM and HTLIC achieve comparable results, while OHTWC achieves the best results. Batch learning algorithm SVM does not have much superiority compared with other online learning algorithms. Remind that in order to fit the online setting, we periodically perform SVM algorithm to train the classifier after receiving $\frac{T}{20}$ instances. SVM algorithm does not have any prior training instances to learn the classifier for the first coming data, which could be the principal reason that SVM does not achieve the lower error rates.

Figure 3 shows detailed learning processes of all used algorithms on several representative classification tasks, and the dotted lines indicate the standard deviations. We see that as the number of target data increases, all the algorithms usually obtain lower error rates. And our proposed OHTWC algorithm

Table 1. Average rate of mistakes on example tasks of text-image data set.

Task	PA	HET	SVM	HTLIC	OHTWC
4	0.3604 ± 0.0163	0.3776 ± 0.0141	0.3395 ± 0.0133	0.3218 ± 0.0103	$\mathbf{0.3075 \pm 0.0090}$
10	0.3668 ± 0.0143	0.4113 ± 0.0210	0.3475 ± 0.0115	0.3514 ± 0.0099	$\mathbf{0.3291 \pm 0.0114}$
11	0.2622 ± 0.0119	0.2947 ± 0.0138	0.2480 ± 0.0138	0.2515 ± 0.0090	$\mathbf{0.2328 \pm 0.0087}$
17	0.2706 ± 0.0100	0.3018 ± 0.0162	0.2462 ± 0.0132	0.2476 ± 0.0166	$\mathbf{0.2235 \pm 0.0089}$
20	0.3015 ± 0.0102	0.3301 ± 0.0134	0.2918 ± 0.0146	0.2907 ± 0.0109	$\mathbf{0.2776 \pm 0.0132}$
22	0.2321 ± 0.0145	0.2635 ± 0.0193	0.2384 ± 0.0136	$\mathbf{0.2043 \pm 0.0095}$	0.2123 ± 0.0110
31	0.4243 ± 0.0132	0.4412 ± 0.0182	0.4316 ± 0.0119	0.4381 ± 0.0184	$\mathbf{0.3547 \pm 0.0102}$
35	0.2298 ± 0.0124	0.2686 ± 0.0102	0.2110 ± 0.0112	0.2421 ± 0.0152	$\mathbf{0.1922 \pm 0.0088}$
41	0.2707 ± 0.0116	0.2838 ± 0.0116	0.2601 ± 0.0126	0.2815 ± 0.0149	$\mathbf{0.2493 \pm 0.0069}$
Average	0.2997	0.3363	0.2844	0.2834	**0.2698**



(a) Task 5 (b) Task 16 (c) Task 35

Fig. 3. Online average rate of mistakes on example tasks of text-image data set.

consistently achieves the best or at least highly competitive results compared with the baseline methods. In addition, OHTWC algorithm usually obtains low mistake rates at the beginning stage, which verifies our approach of heterogeneous transfer does take advantage of useful knowledge from the source domain. Because of the lack of training data, SVM usually gets higher mistake rates, while is able to achieve comparable results by using more training data. Similar results can be observed in other learning tasks.

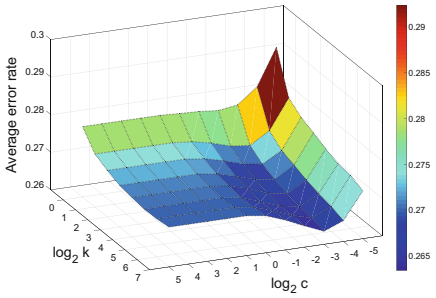


Fig. 4. Results of OHTWC on varying values of parameters c and k.

Parameter Sensitivity. We also investigate how different values of parameters affect the mistake rates of the proposed algorithm. It can be seen that using more nearest neighbors to build an offline classifier can improve the performance of OHTWC algorithm. Nevertheless, the average results do not change too much with respect to parameter c or k. Small numbers of neighbors can also achieve low error rates.

4 Conclusion

In this paper, we propose a novel online heterogeneous transfer learning method, called OHTWC, by leveraging the co-occurrence data of heterogeneous domains. In OHTWC, a heterogeneous similarity via the co-occurrence data is constructed to seek k nearest neighbors (kNN) in the source domain. An offline classifier is built on the source data, while an online classifier is built by using the target data, and we use the Hedge weighting strategy to dynamically combine these two classifiers to make ensemble classification. The theoretical analysis of the proposed OHTWC algorithm is also provided. Experimental results on a real-world data set demonstrate the effectiveness our proposed method.

References

1. Cesa-Bianchi, N., Lugosi, G.: Prediction, Learning, and Games. Cambridge University Press, Cambridge (2006)
2. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines. ACM Trans. Intell. Syst. Technol. **2**(3), 27 (2011)
3. Chua, T.S., Tang, J., Hong, R., Li, H., Luo, Z., Zheng, Y.T.: NUS-WIDE: a real-world web image database from National University of Singapore. In: CIVR (2009)
4. Crammer, K., Dekel, O., Keshet, J., Shalev-Shwartz, S., Singer, Y.: Online passive-aggressive algorithms. J. Mach. Learn. Res. **7**, 551–585 (2006)
5. Dai, W., Chen, Y., Xue, G.R., Yang, Q., Yu, Y.: Translated learning: transfer learning across different feature spaces. In: NIPS (2008)
6. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. J. Comput. Syst. Sci. **55**(1), 119–139 (1997)
7. Ge, L., Gao, J., Zhang, A.: OMS-TL: a framework of online multiple source transfer learning. In: CIKM, pp. 2423–2428 (2013)
8. Hoi, S.C., Wang, J., Zhao, P.: LIBOL: a library for online learning algorithms. J. Mach. Learn. Res. **15**(1), 495–499 (2014)
9. Ng, M.K., Wu, Q., Ye, Y.: Co-transfer learning via joint transition probability graph based method. In: The First International Workshop on Cross Domain Knowledge Discovery in Web and Social Network Mining, KDD (2012)
10. Pan, S.J., Yang, Q.: A survey on transfer learning. IEEE Trans. Knowl. Data Eng. **22**(10), 1345–1359 (2010)
11. Shalev-Shwartz, S., Crammer, K., Dekel, O., Singer, Y.: Online passive-aggressive algorithms. In: NIPS (2004)
12. Tan, B., Song, Y., Zhong, E., Yang, Q.: Transitive transfer learning. In: KDD, pp. 1155–1164 (2015)

13. Tan, B., Zhong, E., Ng, M.K., Yang, Q.: Mixed-transfer: transfer learning over mixed graphs. In: SDM (2014)
14. Wang, B., Pineau, J.: Online boosting algorithms for anytime transfer and multi-task learning. In: AAAI (2015)
15. Wang, G., Hoiem, D., Forsyth, D.: Building text features for object image classification. In: CVPR (2009)
16. Wu, Q., Ng, M.K., Ye, Y.: Cotransfer learning using coupled markov chains with restart. IEEE Intell. Syst. **29**(4), 26–33 (2014)
17. Yang, L., Jing, L., Ng, M.K.: Robust and non-negative collective matrix factorization for text-to-image transfer learning. IEEE Trans. Image Process. **24**(12), 4701–4714 (2015)
18. Yang, L., Jing, L., Yu, J., Ng, M.K.: Learning transferred weights from co-occurrence data for heterogeneous transfer learning. In: IEEE Transactions on Neural Networks and Learning Systems (2015)
19. Zhao, P., Hoi, S.C.: OTL: a framework of online transfer learning. In: ICML (2010)
20. Zhao, P., Hoi, S.C., Wang, J., Li, B.: Online transfer learning. Artif. Intell. **216**, 76–102 (2014)
21. Zhu, Y., Chen, Y., Lu, Z., Pan, S.J., Xue, G.R., Yu, Y., Yang, Q.: Heterogeneous transfer learning for image classification. In: AAAI (2011)