Transfer Heterogeneous Unlabeled Data for Unsupervised Clustering

Shu Kong, Donghui Wang*
Dept. of Computer Science and Technology, Zhejiang University, Hangzhou 310027, PR China
{*aimerykong, dhwang*}*@zju.edu.cn*

Abstract

*In this paper, we propose a novel method called **THUNTER** to **t**ransfer the **h**eterogenous **un**labeled data from the source domain to the target domain for clus**ter**ing. Suppose the target data are a set of images, then the so-called heterogeneous unlabeled data can be a large set of text data or acoustic data. Our method aims to address how to transfer these large amount of heterogeneous data to the relatively smaller target data set for clustering. To the best of our knowledge, it is the first work in the community to transfer the unlabeled data, especially the unlabeled heterogeneous data, for unsupervised clustering. Furthermore, along with our method, a novel **dic**tionary-based data **trans**fer strategy (**DicTrans**) is introduced in this paper, which measures the fidelity of transferring the target data to the source domain and automatically decides how many to transfer. Through a series of experiments, the effectiveness of THUNTER and DicTrans are demonstrated with very promising performances.*

1. Introduction

Clustering is the task of assigning a set of observations into clusters, so that the observations in the same cluster are more similar to each other than to those in other clusters. It is a main task of explorative data mining, and a common technique for statistical data analysis used in many fields, such as machine learning, image analysis, information retrieval, bioinformatics, etc.

Given a set of data, we aim to partition the data into several clusters, *e.g.* clustering the face images from C individuals into C clusters and then allocating the cluster label for a new face image. To achieve satisfactory clustering result, we always expect there are a large amount of data available to guarantee the data intrinsically aggregate together in their own cluster, thus the new data are allocated to the right clusters with high fidelity. But the truth is that, in the real world, no sufficient amount of data are available, *e.g.* we cannot infer the right cluster bounds from limited training set of the face images. How can we obtain a satisfactory clustering result in this situation?

Actually, we are always provided with a number of heterogenous unlabeled data out side the target domain without charge. For example, besides the face images provided by the training set, we can get many text data or acoustic data for free. Intuitively, we may ask can we use these auxiliary data (the source domain) to improve the clustering performance on the face dataset (the target domain), even though the two feature spaces (the dimensionality of the sample data) of the two domains are totally different? Our answer is yes.

To address this problem, we propose a novel method called **THUNTER** to **t**ransfers a subset of **h**eterogenous **un**labeled data (source domain) to the target domain to improve clus**ter**ing performance. In our work, transfer learning plays an important role. There are four approaches used in transfer learning [6]: instance transfer, feature-representation transfer, parameter transfer and relational-knowledge transfer. In the literatures, there are only a few researches on unsupervised transfer learning. A self-taught clustering method is proposed in [2] to learn a common feature space across domains. In [12], transferred discriminative analysis is proposed to use the labeled source data to learn a discriminative subspace for the unlabeled target data. Both of the two methods belong to feature-representation-transfer approach. Whereas our THUNTER aims to select some unlabeled data from heterogenous source set and transfer them to the target set for clustering.

It is widely considered the images of an object *e.g.* facial images of a specific individual, naturally reside on a manifold [10]. But insufficient amount of images will not guarantee the intrinsical aggregation of the data, making the manifolds intermittent and thus the right cluster bounds cannot be inferred. As our model transforms the heterogenous signals into target images and transfers the most informative ones to the target domain,

*This work is supported by 973 Program (No.2010CB327904) and Natural Science Foundations of China (No.61071218).

the defective manifolds will be complemented to some extent, by which way the clustering performance will be improved. To the best of our knowledge, it is the first work to transfer unlabeled data, especially for heterogenous source data, for unsupervised clustering.

The main challenges of our work are to deal with the different feature spaces of the two sets and to determine how and how many source data to transfer. For the first challenge, we hold an assumption that different domains of data are embedded in a lower and common feature subspace, in which the intrinsic similarity among the source and target data are preserved. Hence the key requirement is to find a well-established projection for both the source and target domain. But we do not force the projections from different domains to be totally the same, On the contrary, we learn two projections (for source and target domain) which are bridged by simultaneous dictionary learning process. Although there are some sample selection methods, such as K-L divergence [3, 9] and sample selection bias [7], to address the last two challenges, we introduce another contribution of this paper: the **DicTrans** method based on the learned **dic**tionary to select and **trans**fer the desirable source data. To evaluate the proposed two-step method (THUNTER and DicTrans), we run several experiments across various benchmarks, and the promising performances demonstrate the merit of our methods.

2 The proposed THUNTER — Step I

Denote the target set as $\mathbf{T} = [\mathbf{t}_i] \in \mathbb{R}^{p\times N_T}$, consisting of N_T vector-represented data, and the source set as $\mathbf{S} = [\mathbf{s}_i] \in \mathbb{R}^{q\times N_S}$. What are available to transfer is the heterogeneous source data, therefore the two feature spaces are different, *i.e.* $p \neq q$. To tackle this problem, we can project the source and target data into a common feature subspace, as the method in [8]. Note that the method proposed in [8] has an intrinsic drawback that the number of data in the source and target must be equal, because it has to measure the difference between the two projected data sets. For this reason, it adopts random sampling to increase the smaller set, which will learn a biased cluster centroid and cannot fully represent the true data belonging to this cluster. But our THUNTER do not require this constraint. If the two projections $\mathbf{W}_T \in \mathbb{R}^{p\times k}$ and $\mathbf{W}_S \in \mathbb{R}^{q\times k}$ are acquired for the target and source set, then we can obtain the projected target and source data sets $\mathbf{B}_T = \mathbf{W}_T^T\mathbf{T} \in \mathbb{R}^{k\times N_T}$ and $\mathbf{B}_S = \mathbf{W}_S^T\mathbf{S} \in \mathbb{R}^{k\times N_S}$ respectively (usually $k \leq \min\{p, q\}$). To measure the transferability of the source set, we learn dictionary $\mathbf{D}_T = [\mathbf{d}_{T,i}] \in \mathbb{R}^{k\times d}$ and $\mathbf{D}_S = [\mathbf{d}_{S,i}] \in \mathbb{R}^{k\times d}$ for the target and source projected data set respectively, and use the similarity of the two dictionaries $\|\mathbf{D}_T - \mathbf{D}_S\|_F^2$ to drive the projection matrices to be more desirable. Mathematically, we minimize the following objective function over $\mathbf{W}_T$, $\mathbf{W}_S$, $\mathbf{A}_T$, $\mathbf{A}_S$, $\mathbf{D}_T$ and $\mathbf{D}_S$:

$$\begin{aligned} f =& \alpha\Big(\|\mathbf{W}_T^T\mathbf{T} - \mathbf{D}_T\mathbf{A}_T\|_F^2 + \gamma\Phi(\mathbf{A}_T) \\ &+ \eta\|\mathbf{W}_T\|_F^2\Big) + (1-\alpha)\Big(\|\mathbf{W}_S^T\mathbf{S} - \mathbf{D}_S\mathbf{A}_S\|_F^2 \\ &+ \gamma\Phi(\mathbf{A}_S) + \eta\|\mathbf{W}_S\|_F^2\Big) + \lambda\|\mathbf{D}_T - \mathbf{D}_S\|_F^2 \\ &\text{s.t. } \|\mathbf{d}_{T,j}\|_2 = \|\mathbf{d}_{S,j}\|_2 = 1, \text{ for } \forall j = 1, \ldots, K, \end{aligned} \tag{1}$$

where $\Phi(\cdot)$ is defined as the column-wise ℓ_1-norm penalty, *e.g.* $\Phi(\mathbf{A}_T) = \sum_{i=1}^{N_T}\|\mathbf{a}_{T,i}\|_1$ for $\mathbf{A}_T = [\mathbf{a}_{T,i}] \in \mathbb{R}^{d\times N_T}$. $\alpha \in (0, 1)$ is a regularization parameter balancing the importance of the two sets, and $\lambda > 0$ controls how much similar the two sets are by measuring the two learned dictionaries. γ controls the sparsity of the coefficients, and the Frobenius-norm penalty on $\mathbf{W}_T$ and $\mathbf{W}_S$ makes the solution stable, prohibiting the two projection matrices from being too large. Note that even though there always are a set trivial solutions that $\mathbf{W}_T$, $\mathbf{W}_S$, $\mathbf{A}_T$ and $\mathbf{A}_S$ are zero matrices, owing to some reasonable initializations, we can still obtain the satisfactory non-trivial solutions. Here the term $\|\mathbf{D}_T - \mathbf{D}_S\|_F^2$ is used to bridge the two heterogeneous set and measure the similarity of the projected data sets. It is worth noting that the dictionaries will automatically perform rotation and permutation among their atoms to minimize the difference of them.

Eq. 1 is a joint optimization problem of the projection matrices, the dictionaries and the coefficients. Like many multi-variable optimization problem, we solve it in an alternative manner, the overall procedure is described in the following algorithm.

Algorithm: optimizing objective function Eq. 1

Step 1. Initialize the projections and dictionaries For initializing the projections, we use the leading k left singular vectors of target set $\mathbf{T}$ and source set $\mathbf{S}$ as $\mathbf{W}_T$ and $\mathbf{W}_S$, respectively. Then, we run K-SVD algorithm [1] on the projected target set $\mathbf{B}_T$ and source set $\mathbf{B}_S$ to initialize the dictionary $\mathbf{D}_T$ and $\mathbf{D}_S$. Next, we calculate $\mathbf{A}_T$ and $\mathbf{A}_S$.

Step 2. Updating the coefficients Two coefficient matrices $\mathbf{A}_T$ and $\mathbf{A}_S$ need to be updated, and we update them one by one by fixing other parameters ($\mathbf{W}_T$,$\mathbf{W}_S$,$\mathbf{D}_T$,$\mathbf{D}_S$). Particularly, when updating $\mathbf{A}_T$, we obtain the well-known LASSO problem [11] by ignoring the terms independent of $\mathbf{A}_T$:

$$\mathbf{A}_T = \underset{\mathbf{A}_T}{\operatorname{argmin}} \|\mathbf{W}_T^T\mathbf{T} - \mathbf{D}_T\mathbf{A}_T\|_F^2 + \gamma\Phi(\mathbf{A}_T). \tag{2}$$

Therefore, we can easily solve $\mathbf{A}_T$ by solving this LASSO problem using feature-sign search algorithm [5]. $\mathbf{A}_S$ can be solved in the same way.

Step 3. Updating the dictionaries Two dictionaries need to be updated and they are connected by the term $\|\mathbf{D}_T - \mathbf{D}_S\|_F^2$, hence we update one by fixing the other, and update it atom by atom. By denoting $\mathbf{a}_{T,(j)}$ as the j^{th} row of $\mathbf{A}_T$ and $\bar{\mathbf{T}} = \mathbf{W}_T^T\mathbf{T} - \sum_{i\neq j}\mathbf{d}_{T,i}\mathbf{a}_{T,(i)}$ to update $\mathbf{D}_T$, or more preciously speaking, to update the j^{th} atom $\mathbf{d}_{T,j}$, we can minimize the following over $\mathbf{d}_{T,j}$ by ignoring the unrelated terms:

$$g = \|\bar{\mathbf{T}} - \mathbf{d}_{T,j}\mathbf{a}_{T,(j)}\|_F^2 + \frac{\lambda}{\alpha}\|\mathbf{d}_{T,j} - \mathbf{d}_{S,j}\|_F^2. \quad (3)$$

We can see Eq. 3 is convex on $\mathbf{d}_{T,j}$. By setting the first derivative zero, *i.e.* $\partial g/\partial \mathbf{d}_{T,j} = 0$, we obtain the updated $\mathbf{d}_{T,j} = (\|\mathbf{a}_{T,(j)}\|_2^2 + \frac{\lambda}{\alpha})^{-1}(\bar{\mathbf{T}}\mathbf{a}_{T,(j)}^T + \frac{\lambda}{\alpha}\mathbf{d}_{S,j})$. Note that the atoms of the dictionary should have unit length. Therefore, the final updated atom $\hat{\mathbf{d}}_{T,j} = \mathbf{d}_{T,j}/\|\mathbf{d}_{T,j}\|_2$, as well, the corresponding row in $\mathbf{A}_T$ should be scaled by $\|\mathbf{d}_{T,j}\|_2$, *i.e.* $\hat{\mathbf{a}}_{T,(j)} = \|\mathbf{d}_{T,j}\|_2\mathbf{a}_{T,(j)}$. The source dictionary $\mathbf{D}_S$ can be updated in the same way: $\hat{\mathbf{d}}_{S,j} = \mathbf{d}_{S,j}/\|\mathbf{d}_{S,j}\|_2$ where $\mathbf{d}_{S,j} = (\|\mathbf{a}_{S,(j)}\|_2^2 + \frac{\lambda}{1-\alpha})^{-1}(\bar{\mathbf{S}}\mathbf{a}_{S,(j)}^T + \frac{\lambda}{1-\alpha}\mathbf{d}_{T,j})$ with $\hat{\mathbf{a}}_{S,(j)} = \|\mathbf{d}_{S,j}\|_2\mathbf{a}_{S,(j)}$.

Step 4. Updating the projection matrices As the two projections are independent with each other (although the similarity term on the two dictionaries bridges the projections indirectly), we can obtain closed-form solutions for both of them. When updating $\mathbf{W}_T$, we can simply ignore the unrelated terms and solve the objective as below:

$$\begin{aligned}\mathbf{W}_T &= \underset{\mathbf{W}_T}{\operatorname{argmin}} \|\mathbf{W}_T^T\mathbf{T} - \mathbf{D}_T\mathbf{A}_T\|_F^2 + \eta\|\mathbf{W}_T\|_F^2 \\ &= (\mathbf{T}\mathbf{T}^T + \eta\mathbf{I})^{-1}\mathbf{T}\mathbf{A}_T^T\mathbf{D}_T^T,\end{aligned} \quad (4)$$

where $\mathbf{I}$ is the identity matrix with appropriate size. $\mathbf{W}_S$ can be updated in the same way: $\mathbf{W}_S = (\mathbf{S}\mathbf{S}^T + \eta\mathbf{I})^{-1}\mathbf{S}\mathbf{A}_S^T\mathbf{D}_S^T$.

Step 5. Go back to Step 2 until the values of the objective function f in adjacent iterations are close enough, *i.e.* convergence is achieved. Finally, output $\mathbf{W}_T$, $\mathbf{W}_S$, $\mathbf{A}_T$, $\mathbf{A}_S$, $\mathbf{D}_T$ and $\mathbf{D}_S$. It is straightforward that THUNTER algorithm converges because the updated parameters in each iteration will decrease f.

3 The proposed DicTrans — Step II

Now we present the second step of our model — DicTrans which selects and transfers the most meaningful samples from the source set to the target set. DicTrans uses the learned target dictionary $\mathbf{D}_T$ to represent the samples, and select the ones best represented by $\mathbf{D}_T$, *i.e.* the data to transfer will have less reconstruction error than those are not selected. Note the data used for transferring and clustering are the projected data. Mathematically, we calculate the reconstruction error for every source sample $\mathbf{s}_i$ in a sparse manner (scalar γ controls sparse degree of coefficient $\boldsymbol{\beta}$):

$$e_i = \min_{i} \|\mathbf{W}_S^T\mathbf{s}_i - \mathbf{D}_T\boldsymbol{\beta}_i\|_F^2 + \gamma\|\boldsymbol{\beta}_i\|_1. \quad (5)$$

Moreover, we calculate a threshold $t = \max_i \|\mathbf{W}_T^T\mathbf{t}_i - \mathbf{D}_T\boldsymbol{\beta}_i\|_F^2 + \gamma\|\boldsymbol{\beta}_i\|_1$. Then we transfer the source data whose reconstruction error is less than t. Note that the mixed data set, which consists of the projected target and selected source data, ought to be normalized to unit length for the sequential clustering.

4 Experimental results

Parameters. Throughout this paper, we set $\alpha = 0.5$, $\eta = 0.001$, $\gamma = 0.1$, and $\lambda = 70$. As our work is to study how THUNTER improves the clustering performance, we use the k-means on the projected target data and the selected source data for clustering. As well, k-means is carried out as a baseline on the target set and is compared with THUNTER. For all methods, Euclidean length is the distance metric, and averaged clustering accuracy [4] in 10 runs is reported with the standard deviation. Among the experiments, the number of clusters are set equal to that of actual labels in target set.

Three datasets are used as the source or target set in this paper, they are COIL20[1], ISOLET[2] and ORL database[3]. COIL20 is an image database, containing 20 objects. The images of each objects are taken 5 degrees apart as the object is rotated on a turntable and each object has 72 images. The size of each grayscale image is 32×32 pixels, thus each image is represented by a 1024-dimensional vector. ISOLET is a spoken letter dataset. There are 150 individuals who spoke the name of each letter of the alphabet twice. The speakers are grouped into sets of 30 speakers each, and are referred to as ISOLET1 to ISOLET5. In this paper, we only use ISOLET1 for clustering and instance transferring. The ORL face database contains 400 images, 10 different images for each of 40 distinct individuals. All the images are resized to 30×30-pixel resolution for clustering, thus each image is represented by a 900-dimensional vector. The details of the datasets are summarized in Table 1.

There are 6 sets of experiments for clustering: using one set as the target and either of the other two as the source. The direct comparisons are presented in Table 2. In our experiments, we set the number of atoms (dictionary size when treated as target set) 100

[1] http://www.cs.columbia.edu/CAVE/software/softlib/coil-20.php
[2] http://archive.ics.uci.edu/ml/datasets/ISOLET
[3] http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html

Table 1. Details of the three datasets

Data set	#data	#Feature	#class
COIL20	1440	1024	20
ISOLET	1560	617	26
ORL	400	900	40

Table 2. Clustering Accuracy (%)

Target set	Source set	k-means	**THUNTER**
COIL20	ISOLET	60.78 ± 5.89	**65.64±3.48**
	ORL		**69.35±4.61**
ISOLET	ORL	51.94 ± 4.30	**56.82±3.09**
	COIL20		**54.55±2.17**
ORL	COIL20	56.72 ± 1.68	**68.12±2.36**
	ISOLET		**62.74±3.41**

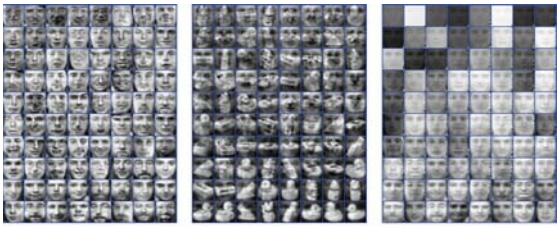

Figure 1. The learned target and source dictionary (best seen by zooming).

for COIL20, 130 for ISOLET and 160 for ORL. Moreover, all the data are projected into $k = 600$ dimensional subspace using $\mathbf{W}_T$ and $\mathbf{W}_S$. As we can see from Table 2, our THUNTER consistently outperforms k-means method. When both of the target and source sets are images, *e.g.* COIL20 and ORL, THUNTER performs much better. Even so, when the source set is audio data, we can still use THUNTER to transfer some projected data to improve clustering performance of the target set. This can be better illustrated by treating ISOLET as the target or source set in Table 2.

Next, to better understand the learned dictionaries, we first resize the images of ORL and COIL20 into 25×25 and fix $\mathbf{W}_T = \mathbf{I}$, and compare the learned $\mathbf{D}_S$ on COIL20 as the source domain with $\mathbf{D}_T$ on ORL. Figure 1 plots three dictionaries: (a) is the learned target dictionary, (b) is the source dictionary with fixed $\mathbf{W}_S = \mathbf{I}$, and (c) is the source dictionary with a learned $\mathbf{W}_S$. Interestingly, some atoms in (b) reflect the silhouettes of the objects in COIL20, and some reflect the faces in ORL. This demonstrates the effectiveness of the term $\|\mathbf{D}_T - \mathbf{D}_S\|_F^2$ to drive the two domain to a similar projected subspace spanned by the two dictionaries. While with learned projection $\mathbf{W}_S$, which projects the COIL20 into a desirable subspace, almost the atoms are delivering the face information. This further manifests the effectiveness of THUNTER to project and transfer the source data, as well as learning the dictionary as similarity measurement of the two domains.

5 Conclusion

In this paper, we introduce a novel method called THUNTER to transform heterogenous unlabeled data of the source domain and the target domain into a common subspace. Then, we develop an approach called DicTrans to automatically select and transfer some informative projected source data to the target set for better clustering performance. Through a series of experiments, the merit of the proposed two-step model (THUNTER and DicTrans) is demonstrated with very promising results.

THUNTER can be easily extended to supervised transfer learning for classification tasks. For example, if we exploit the label information in the source set, we can extend THUNTER to a transductive transfer learning method, in which the intrinsic within-class data structures are carefully considered. Moreover, DicTrans for selecting samples is also need to be refined to seriously take the target data manifold into account. We postpone these researches in our future work.

References

[1] M. Aharon, M. Elad, and A. M. Bruckstein. K-svd: an algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Trans. on Signal Processing*, 2006.

[2] W. Dai, Q. Yang, G. Xue, and Y. Yu. Self-taught clustering. *ICML*, 2008.

[3] Y. Dai, Q. Yang, G.-R. Xue, and Y. Yu. Boosting for transfer learning. *ICML*, 2007.

[4] Q. Gu and J. Zhou. learning the shared subspace for multi-task clustering and transductive transfer classification. *ICDM*, 2009.

[5] H. Lee, A. Battle, R. Raina, and A. Y. Ng. efficient sparse coding algorithms. *NIPS*, 2006.

[6] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Trans. on Knowledge and Data Engineering*, 22(10), 2010.

[7] J. Quionero-Candela, M. Sugiyama, A. Schwaighofer, and N. Lawrence. *Dataset Shift in machine learning*. MIT Press, 2009.

[8] X. Shi, Q. Liu, W. Fan, and P. S. Yu. Transfer across completely different feature spaces via spectral embedding. *IEEE Transactions on Knowledge and Data Engineering*, 2011.

[9] M. Sugiyama, S. Nakajima, H. Kashima, P. Buenau, and M. Kawanabe. direct importance estimation with model selection and its application to covariate shift adaptation. *NIPS*, 2007.

[10] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

[11] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58:267–288, 1994.

[12] Z. Wang, Y. Song, and C. Zhang. Transfer dimensionality reduction. *ECML-PKDD*, 2008.