Transfer learning with part-based ensembles

Shiliang Sun, Zhijie Xu, and Mo Yang

Department of Computer Science and Technology, East China Normal University
500 Dongchuan Road, Shanghai 200241, P. R. China
slsun@cs.ecnu.edu.cn, momo.yang12@gmail.com

Abstract. Transfer learning is one of the most important directions in current machine learning research. In this paper, we propose a new learning framework called Multi-source part-based Transfer Learning (MspbTL), which is one kind of parameter transfer with multiple related source tasks. Dissimilar to many traditional works, we consider how to transfer information from one task to another in the form of integrating transferred information between parts. We regard all the complex tasks as a collection of several constituent parts respectively. It means that transfer learning between two complex tasks can be accomplished by subtransfer learning between their parts. Then, after completing the above information transfer between the source and target tasks, we integrate the models of all the parts in the target task into a whole. Experiments on some real data sets with support vector machines (SVMs) validate the effectiveness of our proposed learning frameworks.

Keywords: Transfer learning, part-based model, multi-source learning, support vector machine.

1 Introduction

Traditional machine learning usually depends on the availability of a large number of data from a single task to train an effective model. However, researchers often confront the situations that there are not enough data available and they have to resort to data from other tasks (source tasks) to aid the learning of the target task. In some cases, even though there are many training examples for the target task, integrating information from other tasks or data sets can still be helpful to improve the performance. Due to the above reasons, some machine learning strategies have been investigated, including multi-task learning [1–3], multi-view learning [4, 5], lifelong learning [6, 7] and transfer learning [8–10]. In this paper, we would like to develop new methods for effective transfer learning.

It should be noted that the target task and source task often have different data distributions in real applications. For example, Wall Street firms often hire physicists to solve finance problems even though there is nothing in common superficially between these two problems [11]. It is easy to see in this example that humans can deal with some problems through applying knowledge learned in one domain to an entirely different one. The reason humans can do this is that they have the ability to choose the essence or the most related part of knowledge which is useful to learn the new task. Nevertheless, computers cannot directly distinguish whether one part in the source task is good or bad to transfer to the target task. As a result, it is significant for us to teach the computer to judge the importance of every part in one complex task. One solution to this problem is to weight different parts differently by their contributions to the task.

Many collections of data exhibit a common underlying structure: they consist of a number of identical parts, each with a range of possible states [12]. Sometimes, although one part of a source task is unsuitable to help learn the target task, there may still exist another part which is useful and helpful for this work. In our paper, to find such parts, we utilize a part-based model. This approach has been used in many fields, especially image processing and computer vision. For instance, Bar-Hillel et al. [13] used this principle to establish the model for the task of object recognition. However, they have just made use of the part-based model in one single task. In this paper, we extend the part-based model to transfer learning. Moreover, for the purpose of avoiding negative transfer [14], which is a situation where knowledge from a source task unexpected deteriorates the performance of target task, the principle of multi-source learning [15] is used in our learning framework as well. Different from the usage of multi-source learning in transfer learning like [16], we will analyze the contribution of every part in every source task to help judge the importance of different parts in the target task. With the help of these principles, we can not only reduce the problem of negative transfer, but also improve the effectiveness of transfer learning.

In this paper, we propose a framework named Multi-source part-based Transfer Learning (Ms-pbTL). It is an extension of pbTL [17], which is a process of parameter transfer using one source task. In pbTL, all the complex tasks are regarded as a collection of constituent parts, and every task can be divided into several parts respectively. This means transfer learning between two complex tasks can be accomplished by sub-transfer learning tasks between their parts. This method is also used in Ms-pbTL.

Some main functions of the frameworks of the Ms-pbTL are described in the following points. Firstly, due to the usage of the part-based model, one task can be divided into a number of parts so as to exploit different latent knowledge. Secondly, the multi-source principle lets us have opportunities to obtain more sets of parameters from different source tasks synchronously, which can be combined in the target task. Finally, we can make a difference not only between different parts in one single source task, but also between the corresponding parts in all the source tasks. This step makes it possible to focus more on the parts which can contribute much more to the target task. Besides these points, in our frameworks, from every part in all the source tasks, we can obtain a set of parameters which can be transferred to its corresponding part in the target task to construct an ensemble of classifiers with support vector machines (SVMs) [18, 19]. At the same time, notice that the parameters about a certain part of one source task can only be transferred into the corresponding part of the target task. Depending on this rule, after reusing all the sets of parameters from the source tasks to help train the ensemble classifiers on their corresponding parts in the target task, we combine these classifiers into a final classifier with weights determined by their accuracy rates. The effectiveness of our proposed learning frameworks is supported by experiments on multiple real data sets.

The remainder of this paper is organized as follows. In Section 2, we describe our framework Ms-pbTL in detail. Then, experiments with our proposed methods are provided in Section 3. Section 4 concludes the whole paper and gives future work directions.

2 Multi-source part-based transfer learning

In this section, we present our transfer learning framework, Ms-pbTL. It is essential for us to pay attention to one special characteristic of this part-based model. According to the part-based principle, the whole will be divided into several parts to learn separately. Different from multi-view learning, some latent relationships exist between every pair of two adjacent parts in the part-based model. For example, one picture of human can be divided into three parts such as head, the upper part of the body and the lower part of the body. Obviously, every part contains many particular features which only exist in the part itself. Nevertheless, serving as the joint of the head and the upper part of the body, the neck is the part which can belong to both of them. As a result, all the features about the neck can be contained in both of these two parts. Consequently, we can summarize the characteristics of the part-based model as follows. On one hand, every part of the whole contains a number of distinctive features which will not belong to other parts. On the other hand, there exist a few features which are used to describe the intersection between two adjacent parts and may appear in both of them.

In order to fully use the benefits of the part-based model in parameter transfer learning, a basic learning framework named part-based Transfer Learning (pbTL) which utilizes one source task was proposed in [17]. For the purpose of avoiding negative transfer and improving the effectiveness of transfer learning, we propose its extended version, a new learning framework, Multi-source part-based Transfer Learning (Ms-pbTL). In the first step, we use SVM to train classifiers and learn a set of optimal parameters for every part of each source task. Then, these sets of parameters need to be transferred to the parts of the target task correspondingly. This is the process of parameter transfer learning. Following this, we can learn several better classifiers trained on the basis of these parameters about every part in the target task and combine them into a final classifier in a weighted fashion at last. The function of this step is to determine which part can contribute much more. Before presenting our learning framework Ms-pbTL, we need to state some key considerations here.

Firstly, we use RBF as the kernel function in the SVM. The detailed formula can be written as:

$$k(x_i, x_j) = e^{-\|x_i - x_j\|^2/2\sigma^2}. \tag{1}$$

Therefore, due to the characteristics of RBF and SVM, the core elements in our parameter transfer learning are the usual regularization coefficient C in the SVM formulation and the width parameter σ in the above (1).

Secondly, we divide the source tasks and target task into several parts in terms of features correspondingly. In this step, we need not only to divide these features into different parts averagely, but also to consider in a characteristic of the part-based model that different parts may be related to each other and contain some common features. For example, according to the supposition that $dimension = 10$ (the dimension of the data set) and $N = 3$ (the number of parts to be divided into), now we can split the first nine features into three parts averagely and supply the last feature for every part so as to create three interdependent four-dimensional parts.

Thirdly, although samples of source and target tasks come from different distributions, we suppose they can be mapped into the same class label set.

The detailed process of Ms-pbTL is given in Table 1.

2.1 Remarks on Ms-pbTL

Through getting the optimal set of parameters of every part in the source task and transferring them to the target task part by part, the merits of different parts can be clearly shown. Moreover, the goal of treating different parts differently can be reached by defining weights as well. The weights can be calculated as:

$$W_{T_i} = \frac{Accuracy_{T_i}}{\sum_{i=1}^{N} Accuracy_{T_i}}. \tag{2}$$

What is more, in (2), to further distinguish the importance of different parts in the target task, we calculate the weights of the classifiers of different parts by the distribution of their accuracy rates on the training data set. These accuracy rates show the percent of samples which are predicted correctly by f_{T_N}. Furthermore, in the output step, we compose the final classifier by the sum of the product of every classifier and its weight. The detailed formula can be written as:

$$h_f(x) = \begin{cases} 1 & \text{if} \quad \sum_{i=1}^{N} W_{T_i} \times f_{T_i}(x) \geq 0 \\ -1 & \text{otherwise.} \end{cases} \tag{3}$$

In Ms-pbTL, we use several source tasks $S_1, \cdots, S_n$ simultaneously to learn the target task. From step 1 to 3, we divide all the source tasks and the target task into N parts. In step 4, due to the fact that now we have n source tasks, we can learn n sets of optimal parameters from them to help every part in the target task to come to n sub-classifiers, respectively. Then, with the help of the accuracy vector acquired in step 5, we can combine all the sub-classifiers about one part into a final sub-classifier in line with step 6. After that, we calculate the weights of each classifier to obtain a final one.

Table 1: Framework of multi-source part-based transfer learning

Input:
set of n source tasks $S_1, \cdots, S_n$ (now each S_i here is a source task) and one target task T, where $S_1, \cdots, S_n$ and T belong to different distributions, but contain the same class label set $Y = \{1, -1\}$ as Os-pbTL.
Initialize the number of parts to be divided.: N
Initialize the parameter set (C, σ) in SVM with RBF.

1. Divide every source task into N parts by their features under the same rule: $\{(S_{11}, \cdots, S_{1N}), \cdots, (S_{n1}, \cdots, S_{nN})\}$.
2. Get N classifiers of every source task on the basis of its parts: $\{(f_{S_{11}}, \cdots, f_{S_{1N}}), \cdots, (f_{S_{n1}}, \cdots, f_{S_{nN}})\}$, and their optimal parameter vectors: $\{[(C_{S_{11}}, \sigma_{S_{11}}), \cdots, (C_{S_{1N}}, \sigma_{S_{1N}})], \cdots, [(C_{S_{n1}}, \sigma_{S_{n1}}), \cdots, (C_{S_{nN}}, \sigma_{S_{nN}})]\}$. (In this paper, we will use cross validation to learn these optimal parameter vectors)
3. Divide the target task into N parts corresponding to the source tasks: $T_1, \cdots, T_N$.
4. According to every part in the target task, we will come to n sub-classifiers: $\{[f_{T_{11}}, \cdots, f_{T_{1n}}], \cdots, [f_{T_{N1}}, \cdots, f_{T_{Nn}}]\}$ by the optimal sets of parameters about the corresponding parts of n source tasks acquired in step 2.
5. Calculate the accuracy rate about every classifier obtained in step 4: $\{[Accuracy_{T_{11}}, \cdots, Accuracy_{T_{1n}}], \cdots, [Accuracy_{T_{N1}}, \cdots, Accuracy_{T_{Nn}}]\}$.
6. Calculate the final classifier of every part of the target task, respectively:for $i = 1, \cdots, N$
   $f_{T_i} = \sum_{j=1}^{n} Accuracy_{T_{ij}} \times f_{T_{ij}}$
   end
7. Calculate the accuracy rate $\{Accuracy_{T_1}, \cdots, Accuracy_{T_N}\}$ of every classifier $\{f_{T_1}, \cdots, f_{T_N}\}$ in the target task.
8. Calculate the weights of $\{f_{T_1}, \cdots, f_{T_N}\}$:
   for $i = 1, \cdots, N$
   $W_{T_i} = \frac{Accuracy_{T_i}}{\sum_{i=1}^{N} Accuracy_{T_i}}$
   end

Output the hypothesis:
$$h_f(x) = \begin{cases} 1 & \text{if} \quad \sum_{i=1}^{N} W_{T_i} \times f_{T_i}(x) \geq 0 \\ -1 & \text{otherwise} \end{cases}$$

In addition to all the description above, it is also important for us to discuss more here. In our paper, we have not considered the problem caused by the diversity between the corresponding parts of the source and target tasks. For example, in step 4, we train the sub-classifiers of every part in the target task on the basis of the optimal sets of parameters learned in the source tasks directly and make no changes. However, sometimes, the sets of parameters learned in the source tasks are not suitable enough to be reused in the target task because of the diversity mentioned above. As a result, if we want to deal with this problem and come to the sets of parameters which are more suitable for the learning of the target task, we can actually just initialize the parameter vectors of every part in the target task by the optimal sets of parameters obtained in the corresponding parts of the source tasks. After that, the work is to update them through continuous iterations with some other processors such as neural networks until coming to the satisfied ones.

3 Experiments

In this section, we implement two groups of experiments. We start with a basic group of experiments with real data sets so as to illustrate the effectiveness of our learning framework. In this group of experiments, for the purpose of implementing our method, Ms-pbTL, we employ two source tasks simultaneously to help learning a target task. Moreover, we do a further study about the influence caused by varying the number of source tasks in Ms-pbTL.

In all these experiments, we set the parameter $N = 3$, which represents the number of parts to be used in the target task and source tasks. Certainly, in practical applications, this number of different tasks can be different and needs to be decided by the characteristics of different tasks. In addition, we compare our method, Ms-pbTL, with basic SVM, transfer learning with basic SVM (Transfer SVM) and pbTL in all the experiments.

3.1 Learning with two source tasks

In this section, we run some experiments on real data sets from UCI repository. Note that all the data sets used here are transformed into binary-class problems. Then, due to the characteristics of different data sets, we use different ways to generate the target task and source tasks and run five sets of experiments on four data sets.

On one hand, data sets $\frac{\text{Segmentation}}{\text{path:cement}}$ and $\frac{\text{Digit}}{5:8}$ are multi-class problems. They are divided into several binary-class sub-data sets by their labels to generate the target task and source tasks. On the other hand, data sets $\frac{\text{Digit}}{3:8}$, German and $\frac{\text{WQ}}{\text{level 5: 7}}$ are binary-class problems, as a result, we need to divide them into several sub-data sets by one specific rule to generate the target task and source tasks. Table 2 provides the summary of the used real data sets.

For each data set in Table 2, we use a specific rule to divide it into the target task and source tasks.

Segmentation is one seven-class data set. We divide the whole data set into several binary-class sub-data sets by their labels to generate the target task and source tasks. We use all the data with label *sky* and *window* as the source task A, the data with label *grass* and $foliage$ as the source task B and the data with label *path* and *cement* as the target task.

Table 2. Summary of data sets

Real data set	$\frac{\text{Segmentation}}{\text{path:cement}}$	$\frac{\text{Digit}}{5:8}$	$\frac{\text{Digit}}{3:8}$	German	$\frac{\text{WQ}}{\text{level 5: 7}}$
Total number of examples	1980	3361	1126	1000	2337
Size of the source task A	$\frac{660}{\text{sky:window}}$	$\frac{1115}{6:2}$	488	230	877
Size of the source task B	$\frac{660}{\text{grass:foliage}}$	$\frac{1134}{3:9}$	330	411	648
Target training set	$\frac{330}{\text{path:cement}}$	$\frac{500}{5:8}$	150	159	500
Target testing set	$\frac{330}{\text{path:cement}}$	$\frac{618}{5:8}$	158	200	312
Dimensions	19	64	64	24	11
Number of classes	6	6	2	2	2

Handwritten Digit is one ten-class data set and here we use two different ways to generate the target task and source tasks. Firstly, similar to data set Segmentation, in $\frac{\text{Digit}}{5:8}$, we use all the data with label 6 and 2 as the source task A, the data with label 3 and 9 as the source task B and the data with label 5 and 8 as the target task.

After that, we get all the data with label 3 and 8 to generate one binary-class data set, $\frac{\text{Digit}}{3:8}$, to run another set of experiments. According to this data set, we divide it into the target task and source tasks on the basis of the value of *dimension six*. All the data according with the rule $dimension\ six < 5$ belong to the source task A, $5 \leq dimension\ six < 10$ belong to the source task B and $dimension\ six \geq 10$ for the target task.

German Credit Data is one binary-class data set. We split the data set on the basis of the feature *Duration of month*. The source task A consists of all the data following the rule $Duration > 24$ while the source task B consists of all the data following the rule $12 < Duration \leq 24$ and $Duration \leq 12$ for the target task.

Wine Quality (WQ) is one eleven-class data set and the assignment of it is to grade the wine quality between 0 to 10. Because the data of different classes are not balanced, we select all the data with label *level* 5 and 7 to generate one binary-class data set which contains 2337 samples. Then, we divide this data set into the target task and source tasks on the basis of the value of feature *Residual sugar*. The source task A consists of all the data following the rule $Residual\ sugar < 3$ while the source task B consists of all the data following the rule $3 \leq Residual\ sugar < 8$ and $Residual\ sugar \geq 8$ for the target task.

Finally, note that we only make use of the source task A to run the experiments of Transfer SVM and pbTL. Furthermore, in our framework Ms-pbTL, due to the fact that we divide all the data sets into three parts by their features randomly, we run the experiments of every data set for ten times and get the mean of them as the final scores. Certainly, standard deviation (Std) will be also calculated synchronously. Moreover, it is significant to demonstrate here that, in order to keep the characteristics of the part-based model, we make most of features be owned by only one part and a few features be shared among all the parts in one task. For example, in our experiments, we make use of the Handwritten Digit data set which contains 64 features. We realign the features randomly at the beginning. Then we divide the first 60 features into three parts averagely and share the rest 4 features for all parts. As a result, we create three interdependent 24-dimensional parts. Table 3 shows the classification results.

Table 3. Accuracy rates of different methods (%)

	SVM	Transfer SVM	pbTL	Ms-pbTL
$\frac{\text{Segmentation}}{\text{path:cement}}$	83.33	87.88	$91.61_{\pm 2.66}$	$94.09_{\pm 2.05}$
$\frac{\text{Digit}}{5:8}$	58.99	49.02	$88.35_{\pm 6.23}$	$93.46_{\pm 1.67}$
$\frac{\text{Digit}}{3:8}$	51.27	56.33	$66.52_{\pm 8.19}$	$80.57_{\pm 7.42}$
German	72.00	74.00	$74.70_{\pm 0.42}$	$75.95_{\pm 0.93}$
$\frac{\text{WQ}}{\text{level 5: 7}}$	69.55	70.19	$74.39_{\pm 2.33}$	$76.47_{\pm 2.54}$

Table 3 shows that pbTL and Ms-pbTL outperform the standard SVM and Transfer SVM in every data set and the results of Ms-pbTL are better than pbTL. In data sets $\frac{\text{Segmentation}}{\text{path:cement}}$, $\frac{\text{Digit}}{5:8}$ and $\frac{\text{Digit}}{3:8}$, our proposed framework improves the results remarkably. Compared with these three data sets, Ms-pbTL makes a less improvement on the data sets German and $\frac{\text{WQ}}{\text{level 5: 7}}$.

What's more, we need to pay attention to the results of $\frac{\text{Digit}}{5:8}$ especially. In the experiments of this data set, though Transfer SVM fails to excel the standard SVM, Ms-pbTL still outperforms standard SVM which illustrates three important points as follows. Firstly, general transfer learning can not exert its benefit all the time. Secondly, even though the whole-based transfer learning has been ineffective, the part-based transfer learning can still be effective. Thirdly, the part-based model can help avoid negative transfer.

Overall speaking, experimental results of real data sets show that the combination of the part-based model and transfer learning can promote the learning efficacy and obtain a higher accuracy with the help of multi-source learning.

3.2 Varying the number of source tasks

Here we intend to study the effect of transfer learning caused by varying the number of source tasks. Our purpose here is to observe the changes of the experimental results about Ms-pbTL with the increase of the number of source tasks. We use two different ways to generate the target task and source tasks.

SCITOS-G5 is one four-class data set which records the wall-following navigation task of one mobile robot. However, because of the sparse of the data of two class, Slight-Right-Turn and Slight-Left-Turn, we just use the data from other two classes, Move-Forward and Sharp-Right-Turn to generate one binary-classes data set to run our experiments. We divide the data set SCITOS-G5 into several parts to generate the target task and source tasks by its first feature, US1, which is the ultrasound sensor at the front of the robot. Details can be seen in Table 4.

Then, for the other set of experiments, in order to acquire enough sub-data sets with different labels to generate the target task and source tasks, we reuse the ten-class data set, Handwritten Digit here. Similar to the experiments of $\frac{\text{Digit}}{5:8}$ in the last part, we come to the target task and source tasks by its class labels as shown in Table 4.

Table 4. Summary of SCITOS-G5 and $\frac{\text{Digit}}{5:8}$

	SCITOS-G5	$\frac{\text{Digit}}{5:8}$
Total number of samples	4302	5620
Dimensions	24	64
Number of classes	2	10
Rule of the target task	$US1 \geq 2.1$	Digit 5 and 8
Training size of the target task	400	500
Testing size of the target task	349	612
Rule of the source task A	$1.5 \leq US1 < 2.1$	Digit 3 and 8
Size of the source task A	615	1126
Rule of the source task B	$1.3 \leq US1 < 1.5$	Digit 5 and 9
Size of the source task B	716	1120
Rule of the source task C	$1.0 \leq US1 < 1.3$	Digit 2 and 6
Size of the source task C	770	1115
Rule of the source task D	$0.8 \leq US1 < 1.0$	Digit 7 and 0
Size of the source task D	619	1120
Rule of the source task E	$US1 < 0.8$	Digit 1 and 4
Size of the source task E	833	1139

Note that we only use the source task A to run the experiments of Transfer SVM and pbTL. Then with the increase of the number of source tasks in Ms-pbTL, we intend to add one more source task into our experiments every time from the source tasks B to E orderly. Furthermore, due to the fact that we divide every data set into three parts randomly by their features, we run the experiments of every data set for ten times and get the mean of them as the final scores. Certainly, standard deviation (Std) will be also calculated synchronously. Detailed results have been given in Table 5. Note that Ms-pbTL, Ms3-pbTL, Ms4-pbTL and Ms5-pbTL represent the results of the experiments about the part-based transfer learning with two, three, four and five source tasks.

Table 5. Accuracy rates of different methods (%)

	SCITOS-G5	$\frac{\text{Digit}}{5:8}$
SVM	81.95	58.99
Transfer SVM	82.52	50.98
pbTL	$84.84_{\pm 1.43}$	$88.97_{\pm 4.06}$
Ms-pbTL	$86.59_{\pm 1.04}$	$94.59_{\pm 1.37}$
Ms3-pbTL	$87.51_{\pm 1.61}$	$95.53_{\pm 1.10}$
Ms4-pbTL	$88.82_{\pm 1.63}$	$95.52_{\pm 1.44}$
Ms5-pbTL	$88.62_{\pm 2.04}$	$95.31_{\pm 1.47}$

According to the results of SCITOS-G5 in Table 5, from Ms-pbTL to Ms4-pbTL, we can see that, as the number of source tasks increases, the increasing degrees of experimental results come to decrease and the negative growth happens to Ms5-pbTL finally. The experiments of $\frac{\text{Digit}}{5:8}$ meet the similar condition as well. The results of this data set reach the peak in Ms3-pbTL and then begin to decrease.

In general, though both Ms-pbTL and Ms3-pbTL perform well, the increasing degrees of experimental results become progressively less obvious with increasing number of source tasks. Therefore, we can derive the following conclusion. Too many source tasks can not lead to a better outcome for transfer learning. The most important point of improving the effectiveness of transfer learning is to select the source tasks which are more similar to the target task rather than use as many source tasks as we can.

4 Conclusion and future work

In this paper, we propose a new learning framework, multi-source part-based transfer learning. From our experiments on real data sets, this framework is proved to be more useful and effective than traditional transfer learning. We conclude the reasons about its feasibility with the following points. Firstly, the part-based model lets us have chance to take advantage of different latent knowledge on one task. Secondly, it also decreases the influence of irrelevant and useless features. Thirdly, the multi-source principle makes us obtain more knowledge from different source tasks to learn the target task. At the same time, it also helps avoid negative transfer.

In the future, how to split one task into several interrelated parts more logically is an interesting direction to study. At the same time, experiments in our paper show that the increase of the number of source tasks does not always improve transfer learning, and therefore it may still be a challenge to study how to select the optimal combination of multiple source tasks to promote transfer learning.

Acknowledgements

This work is supported by the Scientific Research Foundation for the Returned Overseas Chinese Scholars, State Education Ministry, and Shanghai Knowledge Service Platform Project (No. ZF1213).

References

1. Caruana, R.: Multitask Learning. Machine Learning, 28, 41–75 (1997)
2. Dai, W., Yang, Q., Xue, G.R., Yu, Y.: Boosting for transfer learning. In: Proceedings of the 24th International Conference on Machine Learning, pp. 193–200 (2007)
3. Evgeniou, T., Pontil, M.: Regularized multi-task learning. In: Proceedings of 10th International Conference on Knowledge Discovery and Data Mining, pp. 109–117 (2004)
4. Xu, Z., Sun, S.: An algorithm on multi-view adaboost. In: Proceedings of 17th International Conference on Neural Information Processing, pp. 355–362 (2010)
5. Zhang, Q., Sun, S.: Multiple-view multiple-learner active learning. Pattern Recognition, 43(9), 3113–3119 (2010)
6. Thrun, S., Pratt, L.: Learning to Learn. Kluwer Academic Publishers (1997)
7. Thrun, S.: Explanation-Based Neural Network Learning: A Lifelong Learning Approach. Kluwer Academic Publishers, Boston, MA (1996)
8. Cao, B., Liu, N.N., Yang, Q.: Transfer learning for collective link prediction in multiple heterogenous domains. In: Proceedings of the 27th International Conference on Machine Learning, pp. 159–166 (2010)
9. Rohrbach, M., Stark, M., Szarvas, G., Gurevych, I., Schiele, B.: What helps where and why? Semantic relatedness for knowledge transfer. In: Proceedings of the 23rd Computer Society Conference on Computer Vision and Pattern Recognition, pp. 910–917 (2010)
10. Wu, P., Dietterich, T.G.: Improving SVM accuracy by training on auxiliary data sources. In: Proceedings of the 21st International Conference on Machine Learning, pp. 871–878 (2010)
11. Davis, J., Domingos, P.: Deep transfer via second-order Markov logic. In: Proceedings of the 26th International Conference on Machine Learning, pp. 217–224 (2009)
12. Ross, D.A., Zemel, R.S.: Learning parts-based representations of data. Journal of Machine Learning Research, 7, 2369–2397 (2006)
13. Bar-Hillel, A., Hertz, T., Weinshall, D.: Object class recognition by boosting a part-based model. In: Proceedings of IEEE Computer Society Conference on Computer Vision and Recognition, pp. 702–709 (2005)
14. Perkins, D.N., Salomon, G.: Transfer of learning. The Journal of International Encyclopedia of Education, 11, 6452–6457 (1992)

15. Crammer, K., Kearns, M., Wortman, J.: Learning from multiple sources. Journal of Machine Learning Research, 9, 1757–1774 (2008)
16. Luo, P., Zhuang, F., Xiong, Y., He, Q.: Transfer learning from multiple source domains via consensus regularization. In: Proceedings of the 17th ACM Conference on Information and Knowledge Management, pp. 103–112 (2008)
17. Xu, Z., Sun, S.: Part-based transfer learning. In: Proceedings of 8th Internaional Symposium on Neural Networks, pp. 434–441 (2011)
18. Lima, N.H.C., Neto, A.D.D., Melo, J.D.: Creating an ensemble of diverse support vector machines using Adaboost. In: Proceedings of the International Joint Conference on Neural Networks, pp. 2342–2346 (2009)
19. Shawe-Taylor, J., Sun, S.: A review of optimization methodologies in support vector machines. Neurocomputing, 74(17), 3609–3618 (2011)