Transfer Learning

Catarina Oliveira

May 7, 2021

Abstract

This document is a resource for the subject "Estimação, Deteção e Aprendizagem II" presenting the main concepts of Transfer Learning, according to Oliveira (2019), available here.

1 Transfer Learning

Traditional ML and DM methods work under the assumption that the training and testing data are drawn from the same distribution. When the distribution changes, ML models need to be rebuilt from scratch in order to match the new data distribution. This process can be computationally expensive or even impossible if we have large datasets, slow learning processes or if there is no possibility of saving the training data.

There is a need for high-performance learners trained on old data that can be applied to the new data. This can be achieved by transfer learning (TL). TL is inspired in the human ability of reusing learned information (Pan and Yang, 2010). For example, it is easier to recognise pears after learning how to recognise apples. Also, it is easier to learn to play a musical instrument (say, the piano) if one has previous musical knowledge (for example, by knowing how to play the guitar) compared to a person with no musical knowledge at all. Transfer learning aims at producing a model for a target problem with limited training data (or none at all), by exploring knowledge obtained on a different source problem.

1.1 Definition and notation

TL can be characterised by the presence or absence of labelled instances in the source and target domains. In the literature, there is no consensus in the names given to each transfer scenario, when concerning this issue. The same setup is given different names by different authors, as shown on Table 1.

Table 1: Classification of TL mechanisms according to the existence of source and target labelled data.

		Source	
		Present	Absent
Target	Present	Supervised (Chattopadhyay et al., 2012; Daumé III, 2009) Semi-supervised (Blitzer et al., 2006; Gong et al., 2012; Liu et al., 2017) Inductive (Pan and Yang, 2010) Supervised informed (Cook et al., 2013; Feuz and Cook, 2015)	Unsupervised (Pan and Yang, 2010) Unsupervised informed (Cook et al., 2013; Feuz and Cook, 2015)
	Absent	Semi-supervised (Chattopadhyay et al., 2012; Daumé III, 2009) Unsupervised (Blitzer et al., 2006; Gong et al., 2012; Liu et al., 2017) Transductive (Pan and Yang, 2010) Supervised uninformed (Cook et al., 2013; Feuz and Cook, 2015)	Unsupervised (Pan and Yang, 2010) Unsupervised uninformed (Cook et al., 2013; Feuz and Cook, 2015)

In the case we have abundant labelled source data, different names are given to the problem and these are mostly related with the amount of labelled target data: if it is present but limited, some authors name it *supervised* transfer learning (Chattopadhyay et al., 2012; Daumé III, 2009) and others name it *semi-supervised* transfer learning (Blitzer et al., 2006; Gong et al., 2012; Liu et al., 2017); if there is no labelled target data some authors name it *semi-supervised* transfer learning (Chattopadhyay et al., 2012; Daumé III, 2009) and others name it *unsupervised* transfer learning (Blitzer et al., 2006; Gong et al., 2012; Liu et al., 2017).

A different nomenclature is adopted in Pan and Yang (2010), where the authors separate the problems by the existence of labelled source data. If there is none, the problem is called *unsupervised* transfer learning. If labelled source data is present together with some labelled target data, they call it *inductive* transfer learning. Otherwise, if labelled source data is present, but there is no labelled target data, they call it *transductive* transfer learning.

A final example is the nomenclature used by Cook et al. (2013) and Feuz and Cook (2015). In this case, the presence or absence of labelled source data determines the problem to be *supervised* or *unsupervised*, respectively. On the other hand, the presence or absence of labelled target data determines if the problem is *informed* or *uninformed*, respectively. In the remainder of this chapter, we refer to the presence or absence of labelled data on the source and domains instead of using any of the classifications referred above.

To formally define transfer learning, first we will introduce some notation. For consistency, the notation and definition match the ones used in two recent transfer learning surveys (Pan and Yang, 2010; Weiss et al., 2016). For illustration we will continue using the dataset introduced in the beginning of this chapter: a generic dataset containing E instances of I independent variables $x_1, \ldots, x_I$ and one dependent variable y. Thus, x_i^e is the value of the ith independent variable in the eth instance of the dataset.

Notation: A domain $\mathcal{D}$ is defined by two parts: a feature space $\mathcal{X}$ and a marginal probability distribution $P(X)$, where $X = \{x^1, \ldots, x^E\} \in \mathcal{X}$. Considering the generic dataset, x^e is the eth feature vector (instance), E is the number of feature vectors in X, $\mathcal{X}$ is the space of all possible feature vectors, and X is a particular learning sample.

For a given domain $\mathcal{D}$, a task $\mathcal{T}$ is defined by two parts: a label space $\mathcal{Y}$ and a predictive function $f(.)$, which is learned by the feature vector and label pairs $\{x^e, y^e\}$, where $x^e \in X$ and $y^e \in \mathcal{Y}$. Considering the generic dataset, $\mathcal{Y}$ is the set of possible values for the dependent variable, and $f(x)$ is the learner that predicts the label value for the instance x.

From the definitions above, we have a domain $\mathcal{D} = \{\mathcal{X}, P(X)\}$ and task $\mathcal{T} = \{\mathcal{Y}, f(.)\}$. Now, $\mathcal{D}_S$ is defined as the source domain data where $\mathcal{D}_S = \{(x_S^1, y_S^1), \ldots, (x_S^E, y_S^E)\}$, where $x_S^e \in \mathcal{X}$ is the eth data instance of $\mathcal{D}_S$ and $y_S^e \in \mathcal{Y}$ is the corresponding label for x_S^e. In the same way, $\mathcal{D}_T$ is defined as the target domain data where $\mathcal{D}_T = \{(x_T^1, y_T^1), \ldots, (x_T^E, y_T^E)\}$, where $x_T^e \in \mathcal{X}$ is the eth data instance of $\mathcal{D}_T$ and $y_T^e \in \mathcal{Y}$ is the corresponding label for x_T^e.

Furthermore, the source task is denoted as $\mathcal{T}_S$, the target task as $\mathcal{T}_T$, the source predictive function as $f_S(.)$, and the target predictive function as $f_T(.)$.

Definition: *Given a source domain $\mathcal{D}_S$ and a learning task $\mathcal{T}_S$, a target domain $\mathcal{D}_T$ and a learning task $\mathcal{T}_T$, transfer learning aims to help improve the learning of the target predictive function $f_T(.)$ in $\mathcal{D}_T$ using the knowledge in $\mathcal{D}_S$ and $\mathcal{T}_S$, where $\mathcal{D}_S \neq \mathcal{D}_T$, or $\mathcal{T}_S \neq \mathcal{T}_T$.*

Given the notation and definition we will now discuss the situations in which transfer learning can occur. A domain can be defined as $\mathcal{D} = \{\mathcal{X}, P(X)\}$ and a task can be defined as $\mathcal{T} = \{\mathcal{Y}, f(.)\}$, which is the same as $\mathcal{T} = \{\mathcal{Y}, P(Y|X)\}$. Therefore, we have that $\mathcal{D}_S = \{\mathcal{X}_S, P(X_S)\}$ and $\mathcal{T}_S = \{\mathcal{Y}_S, P(Y_S|X_S)\}$ for the source problem. The same happens for the target problem: $\mathcal{D}_T = \{\mathcal{X}_T, P(X_T)\}$ and $\mathcal{T}_T = \{\mathcal{Y}_T, P(Y_T|X_T)\}$. This way, transfer learning can occur when we have at least one of the following situations:

- $\mathcal{X}_S \neq \mathcal{X}_T$: the domains' feature spaces are different. This is claaed *heterogeneous transfer learning* (Day and Khoshgoftaar, 2017) and its most common approach consists in aligning the feature spaces. Similarly, when the feature spaces are the same ($\mathcal{X}_S = \mathcal{X}_T$) it is called *homogeneous transfer learning*. It usually aims at reducing distribution differences.

- $P(X_S) \neq P(X_T)$: this happens when the domains have the same features, but their marginal distributions are different (e.g., different frequencies in domain-specific features). A common approach in this case is domain adaptation, which consists in altering a source domain trying to make its distribution closer to the target's.

- $\mathcal{Y}_S \neq \mathcal{Y}_T$: there is a mismatch in the class space (e.g. different number of classes in the source and target problems).

- $P(Y_S|X_S) \neq P(Y_T|X_T)$: the conditional probability distribution of the source and target domains are different. This happens, for example, when the same feature value has two different meanings on the source and target domains.

There are three issues to take into account in transfer learning: *what*, *how* and *when* to transfer (Pan and Yang, 2010). The first question, *what to transfer?*, concerns the type of information transferred between the problems. The question *how to transfer?* concerns the algorithms used for the transfer of information between problems. The last question, *when to transfer?*, means to know in which situations the transfer should be performed.

1.2 *What* and *how* to transfer?

The first two questions (*what to transfer?* and *how to transfer?*) are closely related. Next, we will categorise the TL mechanisms in terms of the type of information transferred between problems (*what to transfer?*) while, at the same time, we will present algorithms used for the transfer of information between problems (*how to transfer?*). At the end of this subsection (Table 2), we present a summary of this TL categorisation. The transferred information belongs to one of four categories – instances, parameters, relational knowledge or features:

1. **Instance transfer** occurs when instances from the source domain are used for training the model for the target domain. This type of transfer occurs mostly on homogeneous TL scenarios. For example, the algorithm TrAdaBoost (Dai et al., 2007b) uses parts of the labelled train data (source) that have the same distribution as the test data (target) to help constructing the target classification model. Also, the algorithm *kernel mean matching (KMM)* (Huang et al., 2007) tries to match distributions in source and target feature spaces. Another example is the algorithm *Kullback-Leibler Importance Estimation Procedure (KLIEP)* (Sugiyama et al., 2008) that uses the Kullback-Leibler divergence to find important instances to be transferred from the source to the target problem. In Liu et al. (2018) an ensemble framework (TrResampling) is proposed to transfer instances for classification tasks.

2. **Parameter transfer** occurs when the source and target learners share parameters or when ensemble learners are created by combining multiple source learners to form an improved target learner. Approaches to this type of transfer include weighting several source models according to target characteristics (Gao et al., 2008), from within a group of classifiers finding the source classifier that minimizes the error on the target (that happens in Yao and Doretto (2010) in algorithms MultiSource TrAdaBoost – that handles the conditional distribution differences between domains – and TaskTrAdaBoost), weighted training with source data to predict target pseudo-labels and with all this information then predict the target final labels. This is the case of algorithms *Conditional Probability based Multi-source Domain Adaptation (CP-MDA)* (Chattopadhyay et al., 2012) and *Domain Selection Machine (DSM)* (Duan et al., 2012b). These algorithms handle both marginal and conditional distribution differences between the domains. Finally, another approach is to directly transfer the parameters between problems. This is the case in algorithm *Multi Model Knowledge Transfer (MMKT)* (Tommasi et al., 2010), that handles the conditional distribution differences between domains.

3. **Relational knowledge transfer** occurs when the transferred knowledge is based on some relationship between the source and target domains. This is the least used approach in TL. There are some examples of this type of transfer in the literature. Algorithm *Deep Transfer via Markov logic (DTM)* (Davis and Domingos, 2009) discovers structural regularities in the source and instantiates them with predicates from the target problem. Another example is the algorithm *Relational Adaptive bootstraPping (RAP)* (Li et al., 2012), which uses sentiment words

as a link between source and target domains and iteratively builds a target classifier from the two domains by scoring sentence structure patterns, while trying to avoid the marginal distribution differences between the domains. In Xiong et al. (2018), models are transferred to improve anomaly detection. In another approach (Saeedi et al., 2016) the authors transfer data mapping between sensors.

4. **Feature transfer** occurs when features are transferred across domains. This type of transfer is the most used when dealing with heterogeneous TL settings. Feature transfer can be defined as symmetric or asymmetric (Weiss et al., 2016):

   (a) In **symmetric** feature transfer, a common latent feature space between the domains is discovered.

      i. For **homogeneous** TL problems, usually the aim is to overcome the marginal distribution differences among the domains. This can be achieved by discovering a set of latent features between the source and target problems, as in the algorithms *Domain Adaptation of Sentiment classifiers (DAS)* (Glorot et al., 2011) and *Transfer Component Analysis (TCA)* (Pan et al., 2011). Other approaches include finding correspondences between features (Wang and Mahadevan, 2008), learn feature representations by modelling co-occurrence between domain-independent and domain-specific features (as in algorithm *Spectral Feature Alignment (SFA)* (Pan et al., 2010)), or finding domain-independent features (as in algorithm *geodesic flow kernel (GFK)* (Gong et al., 2012)).

      ii. For **heterogeneous** TL problems the most usual approaches are discovering common features, clustering and feature augmentation. In the first technique, the algorithms find common sets of (present or latent) features between the domains. The target model is trained with the source data and applied to the target problem. This happens, for example, in Blitzer et al. (2007), Blitzer et al. (2008), Pan et al. (2008), and Raina et al. (2007) and also in the algorithms *Structural Correspondence Learning (SCL)* (Blitzer et al., 2006), *Topic-bridged probabilistic semantic analysis (TPLSA)* (Xue et al., 2008), *Heterogeneous Spectral Mapping (HeMap)* (Shi et al., 2010), *Translator of Text to Images (TTI)* (Qi et al., 2011), *Domain Adaptation using Manifold Alignment (DAMA)* (Wang and Mahadevan, 2011) and *Heterogeneous Transfer Learning for Text Classification (HTLIC)* (Zhu et al., 2011). The clustering technique consists in clustering source and target data simultaneously to infer common structures between the domains. This is the case in the algorithms *Co-clustering based classification (CoCC)* (Dai et al., 2007a), *Self-taught clustering (STC)* (Dai et al., 2008) and *Transfer Discriminative Analysis (TDA)* (Wang et al., 2008) The feature augmentation technique consists in adding target and common features to the source feature set. This technique is implemented in algorithms *Heterogeneous feature adaptation (HFA)* (Duan et al., 2012c) and *Semi-supervised HFA (SHFA)* (Li et al., 2014). Other approaches include modelling the relevance of features by using metafeatures (Lee et al., 2007) and use manually paired sets of features to be transferred. This last approach is used for example in the algorithm *Cross-Language Text Classification using Structural Correspondence Learning (CL-SCL)* (Prettenhofer and Stein, 2010) by translating words from English to other languages to be able to use the models created for texts written in English to classify texts in other languages.

   (b) In **asymmetric** feature transfer, the source features are re-weighted to resemble the target features.

      i. For **homogeneous** TL problems, the most common approach is to first learn target pseudo-labels by using the source problem for training and then using the pseudo-labels to learn the final target labels. This technique can be used to approximate the domains marginal (as happens in the *Domain Transfer Multiple Kernell Learner (DTMKL)* (Duan et al., 2012a)) or conditional distribution (as in the *Feature Augmentation Method (FAM)* (Daumé III, 2009)), and even both (as is the case of the algorithm *Joint Distribution Adaptation (JDA)* (Long et al., 2013)).

      ii. For **heterogeneous** TL problems, usually a transformation from the source to the target is found. This happens in *Multiple Outlook MAPping (MOMAP)* (Harel and

Mannor, 2010), *Asymmetric Regularized cross-domain transformation (ARC-t)* (Kulis et al., 2011), *Sparse Heterogeneous Feature Representation (SHFR)* (Zhou et al., 2014b) and *Hybrid Heterogeneous Transfer learning (HHTL)* (Zhou et al., 2014a). Another approach consists in training the target model on a set of similar source features. This is the case of the algorithm *Heterogeneous Feature Prediction (HFP)* (Nam et al., 2017), where the similarity of features is obtained by a Kolmogorov-Smirnov test.

Table 2 contains a summary of the referred TL algorithms, considering *what* and *how* to transfer. Since the problems considered on the algorithms described do not match our problems, instead of reusing one of the referred algorithms, we create a weight transfer algorithm described later on Subsection **??**.

Table 2: Summary of transfer learning algorithms.

Instance Transfer	**Parameter Transfer**	**Rel. Knw. Transfer**
KMM (Huang et al., 2007)	CP-MDA (Chattopadhyay et al., 2012)	DTM (Davis and Domingos, 2009)
KLIEP (Sugiyama et al., 2008)	DSM (Duan et al., 2012b)	RAP (Li et al., 2012)
	MMKT (Tommasi et al., 2010)	

Feature Transfer			
	Homogeneous	Heterogeneous	
Symmetric	DAS (Glorot et al., 2011)	SCL (Blitzer et al., 2006)	CoCC (Dai et al., 2007a)
	TCA (Pan et al., 2011)	TPLSA (Xue et al., 2008)	STC (Dai et al., 2008)
	SFA (Pan et al., 2010)	HeMap (Shi et al., 2010)	TDA (Wang et al., 2008)
	GFK (Gong et al., 2012)	TTI (Qi et al., 2011)	HFA (Duan et al., 2012c)
		DAMA (Wang and Mahadevan, 2011)	SHFA (Li et al., 2014)
		HTLIC (Zhu et al., 2011)	CL-SCL (Prettenhofer and Stein, 2010)
Asymm.	DTMKL (Duan et al., 2012a)	MOMAP (Harel and Mannor, 2010)	HHTL (Zhou et al., 2014a)
	FAM (Daumé III, 2009)	ARC-t (Kulis et al., 2011)	HFP (Nam et al., 2017)
	JDA (Long et al., 2013)	SHFR (Zhou et al., 2014b)	

1.3 *When* to transfer?

The ultimate objective of knowing *when to transfer* is to avoid *negative transfer*: when the transfer can harm the learning process in the target task. This issue is referred in Rosenstein et al. (2005), where the authors wish to identify when transfer learning will hurt the performance of the algorithm instead of improving it.

In the literature, there are several approaches used to try to avoid negative transfer, for example:

- Measuring data relatedness, group (or cluster) the several tasks at hand, and then only transfer between tasks that belong to the same group (Bakker and Heskes, 2003; Ben-David and Schuller, 2003; Argyriou et al., 2008; Ge et al., 2014);
- Selecting a limited amount of target data to be labelled (Liao et al., 2005);
- Removing misleading source instances (Jiang and Zhai, 2007; Ngiam et al., 2018);
- Accounting for measures that illustrate the gain in transferring, like *trade-off of transferring* (Blitzer et al., 2008), *transferability* (Eaton et al., 2008) or *PDM: Predictive Distribution Matching* (Seah et al., 2013);
- Choosing only some of the source data to be transferred (Mahmud and Ray, 2008) or just proper subsets of common features (Wang et al., 2008);
- Weight the transferred information, such that the most related sources have higher weights (Tommasi et al., 2010), which can be extended by also weighting the instances to be transferred (Yao and Doretto, 2010);
- Selecting only the most relevant domains (Duan et al., 2012b), which can be done by using specific metrics, as is the example of *ROD: Rank of Domain* (Gong et al., 2012) to evaluate which source domain to choose for transfer.

In our work, we aim to use MtL to help preventing negative transfer. This way, instead of reusing the referred metrics, we generate metafeatures that will be used on the MtL process to try to predict when the transfer will have a positive impact.

References

Argyriou, A., Maurer, A., Pontil, M., 2008. An algorithm for transfer learning in a heterogeneous environment, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer. pp. 71–85. doi:10.1007/978-3-540-87479-9_23.

Bakker, B., Heskes, T., 2003. Task clustering and gating for bayesian multitask learning. Journal of Machine Learning Research 4, 83–99. doi:10.1162/153244304322765658.

Ben-David, S., Schuller, R., 2003. Exploiting task relatedness for multiple task learning, in: Learning Theory and Kernel Machines. Springer, pp. 567–580. doi:10.1007/978-3-540-45167-9_41.

Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., Wortman, J., 2008. Learning bounds for domain adaptation, in: Advances in neural information processing systems, pp. 129–136. URL: https://dl.acm.org/doi/10.5555/2981562.2981579.

Blitzer, J., Dredze, M., Pereira, F., 2007. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification, in: Proceedings of the 45th annual meeting of the association of computational linguistics, pp. 440–447. URL: https://www.aclweb.org/anthology/P07-1056.pdf.

Blitzer, J., McDonald, R., Pereira, F., 2006. Domain adaptation with structural correspondence learning, in: Proceedings of the 2006 conference on empirical methods in natural language processing, Association for Computational Linguistics. pp. 120–128. URL: https://www.aclweb.org/anthology/W06-1615.pdf.

Chattopadhyay, R., Sun, Q., Fan, W., Davidson, I., Panchanathan, S., Ye, J., 2012. Multisource domain adaptation and its application to early detection of fatigue. ACM Transactions on Knowledge Discovery from Data (TKDD) 6, 18. doi:10.1145/2382577.2382582.

Cook, D., Feuz, K.D., Krishnan, N.C., 2013. Transfer learning for activity recognition: A survey. Knowledge and information systems 36, 537–556. doi:10.1007/s10115-013-0665-3.

Dai, W., Xue, G.R., Yang, Q., Yu, Y., 2007a. Co-clustering based classification for out-of-domain documents, in: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM. pp. 210–219. doi:10.1145/1281192.1281218.

Dai, W., Yang, Q., Xue, G.R., Yu, Y., 2007b. Boosting for transfer learning, in: Proceedings of the 24th international conference on Machine learning, ACM. pp. 193–200. doi:10.1145/1273496.1273521.

Dai, W., Yang, Q., Xue, G.R., Yu, Y., 2008. Self-taught clustering, in: Proceedings of the 25th international conference on Machine learning, ACM. pp. 200–207. doi:10.1145/1390156.1390182.

Daumé III, H., 2009. Frustratingly easy domain adaptation. arXiv preprint arXiv:0907.1815 URL: https://arxiv.org/abs/0907.1815.

Davis, J., Domingos, P., 2009. Deep transfer via second-order markov logic, in: Proceedings of the 26th annual international conference on machine learning, ACM. pp. 217–224. doi:10.1145/1553374.1553402.

Day, O., Khoshgoftaar, T.M., 2017. A survey on heterogeneous transfer learning. Journal of Big Data 4, 29. doi:10.1186/s40537-017-0089-0.

Duan, L., Tsang, I.W., Xu, D., 2012a. Domain transfer multiple kernel learning. IEEE Transactions on Pattern Analysis and Machine Intelligence 34, 465–479. doi:10.1109/TPAMI.2011.114.

Duan, L., Xu, D., Chang, S.F., 2012b. Exploiting web images for event recognition in consumer videos: A multiple source domain adaptation approach, in: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE. pp. 1338–1345. doi:10.1109/CVPR.2012.6247819.

Duan, L., Xu, D., Tsang, I., 2012c. Learning with augmented features for heterogeneous domain adaptation. arXiv preprint arXiv:1206.4660 URL: https://arxiv.org/ftp/arxiv/papers/1206/1206.4660.pdf.

Eaton, E., Lane, T., et al., 2008. Modeling transfer relationships between learning tasks for improved inductive transfer, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer. pp. 317–332. doi:10.1007/978-3-540-87479-9_39.

Feuz, K.D., Cook, D.J., 2015. Transfer learning across feature-rich heterogeneous feature spaces via feature-space remapping (fsr). ACM Transactions on Intelligent Systems and Technology (TIST) 6, 3. doi:10.1145/2629528.

Gao, J., Fan, W., Jiang, J., Han, J., 2008. Knowledge transfer via multiple model local structure mapping, in: Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM. pp. 283–291. doi:10.1145/1401890.1401928.

Ge, L., Gao, J., Ngo, H., Li, K., Zhang, A., 2014. On handling negative transfer and imbalanced distributions in multiple source transfer learning. Statistical Analysis and Data Mining: The ASA Data Science Journal 7, 254–271. doi:10.1002/sam.11217.

Glorot, X., Bordes, A., Bengio, Y., 2011. Domain adaptation for large-scale sentiment classification: A deep learning approach, in: Proceedings of the 28th international conference on machine learning (ICML-11), pp. 513–520. doi:10.5555/3104482.3104547.

Gong, B., Shi, Y., Sha, F., Grauman, K., 2012. Geodesic flow kernel for unsupervised domain adaptation, in: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE. pp. 2066–2073. doi:10.1109/CVPR.2012.6247911.

Harel, M., Mannor, S., 2010. Learning from multiple outlooks. arXiv preprint arXiv:1005.0027 URL: https://arxiv.org/abs/1005.0027.

Huang, J., Gretton, A., Borgwardt, K.M., Schölkopf, B., Smola, A.J., 2007. Correcting sample selection bias by unlabeled data, in: Advances in neural information processing systems, pp. 601–608. URL: https://proceedings.neurips.cc/paper/2006/hash/a2186aa7c086b46ad4e8bf81e2a3a19b-Abstract.html.

Jiang, J., Zhai, C., 2007. Instance weighting for domain adaptation in nlp, in: Proceedings of the 45th annual meeting of the association of computational linguistics, pp. 264–271. doi:https://www.aclweb.org/anthology/P07-1034.pdf.

Kulis, B., Saenko, K., Darrell, T., 2011. What you saw is not what you get: Domain adaptation using asymmetric kernel transforms, in: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, IEEE. pp. 1785–1792. doi:10.1109/CVPR.2011.5995702.

Lee, S.I., Chatalbashev, V., Vickrey, D., Koller, D., 2007. Learning a meta-level prior for feature relevance from multiple related tasks, in: Proceedings of the 24th international conference on Machine learning, ACM. pp. 489–496. doi:10.1145/1273496.1273558.

Li, F., Pan, S.J., Jin, O., Yang, Q., Zhu, X., 2012. Cross-domain co-extraction of sentiment and topic lexicons, in: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1, Association for Computational Linguistics. pp. 410–419. doi:10.5555/2390524.2390582.

Li, W., Duan, L., Xu, D., Tsang, I.W., 2014. Learning with augmented features for supervised and semi-supervised heterogeneous domain adaptation. IEEE transactions on pattern analysis and machine intelligence 36, 1134–1148. doi:10.1109/TPAMI.2013.167.

Liao, X., Xue, Y., Carin, L., 2005. Logistic regression with an auxiliary data source, in: Proceedings of the 22nd international conference on Machine learning, ACM. pp. 505–512. doi:10.1145/1102351.1102415.

Liu, F., Zhang, G., Lu, H., Lu, J., 2017. Heterogeneous unsupervised cross-domain transfer learning. arxiv preprint. arXiv preprint arXiv:1701.02511 doi:10.1109/TNNLS.2020.2973293.

Liu, X., Liu, Z., Wang, G., Cai, Z., Zhang, H., 2018. Ensemble transfer learning algorithm. IEEE Access 6, 2389–2396. doi:10.1109/ACCESS.2017.2782884.

Long, M., Wang, J., Ding, G., Sun, J., Yu, P.S., 2013. Transfer feature learning with joint distribution adaptation, in: Proceedings of the IEEE international conference on computer vision, pp. 2200–2207. doi:10.1109/ICCV.2013.274.

Mahmud, M., Ray, S., 2008. Transfer learning using kolmogorov complexity: Basic theory and empirical evaluations, in: Advances in neural information processing systems, pp. 985–992. doi:10.5555/2981562.2981686.

Nam, J., Fu, W., Kim, S., Menzies, T., Tan, L., 2017. Heterogeneous defect prediction. IEEE Transactions on Software Engineering doi:10.1109/TSE.2017.2720603.

Ngiam, J., Peng, D., Vasudevan, V., Kornblith, S., Le, Q.V., Pang, R., 2018. Domain adaptive transfer learning with specialist models. arXiv preprint arXiv:1811.07056 URL: https://arxiv.org/abs/1811.07056.

Oliveira, C.F., 2019. Metalearning for multiple-domain transfer learning URL: https://repositorio-aberto.up.pt/bitstream/10216/120770/2/338468.pdf.

Pan, S.J., Kwok, J.T., Yang, Q., 2008. Transfer learning via dimensionality reduction., in: AAAI, pp. 677–682. URL: https://www.aaai.org/Papers/AAAI/2008/AAAI08-108.pdf.

Pan, S.J., Ni, X., Sun, J.T., Yang, Q., Chen, Z., 2010. Cross-domain sentiment classification via spectral feature alignment, in: Proceedings of the 19th international conference on World wide web, ACM. pp. 751–760. URL: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.181.20&rep=rep1&type=pdf.

Pan, S.J., Tsang, I.W., Kwok, J.T., Yang, Q., 2011. Domain adaptation via transfer component analysis. IEEE Transactions on Neural Networks 22, 199–210. doi:10.1109/TNN.2010.2091281.

Pan, S.J., Yang, Q., 2010. A survey on transfer learning. IEEE Transactions on knowledge and data engineering 22, 1345–1359. doi:10.1109/TKDE.2009.191.

Prettenhofer, P., Stein, B., 2010. Cross-language text classification using structural correspondence learning, in: Proceedings of the 48th annual meeting of the association for computational linguistics, Association for Computational Linguistics. pp. 1118–1127. URL: https://www.aclweb.org/anthology/P10-1114.pdf.

Qi, G.J., Aggarwal, C., Huang, T., 2011. Towards semantic knowledge propagation from text corpus to web images, in: Proceedings of the 20th international conference on World wide web, ACM. pp. 297–306. doi:10.1145/1963405.1963449.

Raina, R., Battle, A., Lee, H., Packer, B., Ng, A.Y., 2007. Self-taught learning: transfer learning from unlabeled data, in: Proceedings of the 24th international conference on Machine learning, ACM. pp. 759–766. doi:10.1145/1273496.1273592.

Rosenstein, M.T., Marx, Z., Kaelbling, L.P., Dietterich, T.G., 2005. To transfer or not to transfer, in: NIPS 2005 workshop on transfer learning, pp. 1–4. URL: http://web.engr.oregonstate.edu/~tgd/publications/rosenstein-marx-kaelbling-dietterich-hnb-nips2005-transfer-workshop.pdf.

Saeedi, R., Ghasemzadeh, H., Gebremedhin, A.H., 2016. Transfer learning algorithms for autonomous reconfiguration of wearable systems, in: 2016 IEEE International Conference on Big Data (Big Data), IEEE. pp. 563–569. doi:10.1109/BigData.2016.7840648.

Seah, C.W., Ong, Y.S., Tsang, I.W., 2013. Combating negative transfer from predictive distribution differences. IEEE transactions on cybernetics 43, 1153–1165. doi:0.1109/TSMCB.2012.2225102.

Shi, X., Liu, Q., Fan, W., Philip, S.Y., Zhu, R., 2010. Transfer learning on heterogenous feature spaces via spectral transformation, in: Data Mining (ICDM), 2010 IEEE 10th International Conference on, IEEE. pp. 1049–1054. doi:10.1109/ICDM.2010.65.

Sugiyama, M., Nakajima, S., Kashima, H., Buenau, P.V., Kawanabe, M., 2008. Direct importance estimation with model selection and its application to covariate shift adaptation, in: Advances in neural information processing systems, pp. 1433–1440. URL: https://dl.acm.org/doi/10.5555/2981562.2981742.

Tommasi, T., Orabona, F., Caputo, B., 2010. Safety in numbers: Learning categories from few examples with multi model knowledge transfer, in: Proceedings of IEEE Computer Vision and Pattern Recognition Conference. doi:10.1109/CVPR.2010.5540064.

Wang, C., Mahadevan, S., 2008. Manifold alignment using procrustes analysis, in: Proceedings of the 25th international conference on Machine learning, ACM. pp. 1120–1127. doi:10.1145/1390156.1390297.

Wang, C., Mahadevan, S., 2011. Heterogeneous domain adaptation using manifold alignment, in: IJCAI proceedings-international joint conference on artificial intelligence, p. 1541. doi:10.5591/978-1-57735-516-8/IJCAI11-259.

Wang, Z., Song, Y., Zhang, C., 2008. Transferred dimensionality reduction, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer. pp. 550–565. doi:10.1007/978-3-540-87481-2_36.

Weiss, K., Khoshgoftaar, T.M., Wang, D., 2016. A survey of transfer learning. Journal of Big Data 3, 9. doi:10.1186/s40537-016-0043-6.

Xiong, P., Zhu, Y., Sun, Z., Cao, Z., Wang, M., Zheng, Y., Hou, J., Huang, T., Que, Z., 2018. Application of transfer learning in continuous time series for anomaly detection in commercial aircraft flight data, in: 2018 IEEE International Conference on Smart Cloud (SmartCloud), IEEE. pp. 13–18. doi:10.1109/SmartCloud.2018.00011.

Xue, G.R., Dai, W., Yang, Q., Yu, Y., 2008. Topic-bridged plsa for cross-domain text classification, in: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval, ACM. pp. 627–634. doi:10.1145/1390334.1390441.

Yao, Y., Doretto, G., 2010. Boosting for transfer learning with multiple sources, in: Computer vision and pattern recognition (CVPR), 2010 IEEE conference on, IEEE. pp. 1855–1862. doi:10.1109/CVPR.2010.5539857.

Zhou, J.T., Pan, S.J., Tsang, I.W., Yan, Y., 2014a. Hybrid heterogeneous transfer learning through deep learning., in: AAAI, pp. 2213–2220. URL: https://dl.acm.org/doi/10.5555/2892753.2892859.

Zhou, J.T., Tsang, I.W., Pan, S.J., Tan, M., 2014b. Heterogeneous domain adaptation for multiple classes, in: Artificial Intelligence and Statistics, pp. 1095–1103. URL: http://proceedings.mlr.press/v33/zhou14.html.

Zhu, Y., Chen, Y., Lu, Z., Pan, S.J., Xue, G.R., Yu, Y., Yang, Q., 2011. Heterogeneous transfer learning for image classification., in: AAAI. URL: https://dl.acm.org/doi/10.5555/2900423.2900630.