Domain Generalization based on Transfer Component Analysis

Thomas Grubinger[1], Adriana Birlutiu[1,2] Holger Schöner[1], Thomas Natschläger[1], and Tom Heskes[3]

[1] Data Analysis Systems, Software Competence Center Hagenberg, Austria
[2] Faculty of Science, "1 Decembrie 1918" University of Alba-Iulia, Romania
[3] Institute for Computing and Information Sciences, Radboud University Nijmegen, Netherlands

Abstract. This paper investigates domain generalization: How to use knowledge acquired from related domains and apply it to new domains? Transfer Component Analysis (TCA) learns a shared subspace by minimizing the dissimilarities across domains, while maximally preserving the data variance. We propose Multi-TCA, an extension of TCA to multiple domains as well as Multi-SSTCA, which is an extension of TCA for semi-supervised learning. In addition to the original application of TCA for domain adaptation problems, we show that Multi-TCA can also be applied for domain generalization. Multi-TCA and Multi-SSTCA are evaluated on two publicly available datasets with the tasks of landmine detection and Parkinson telemonitoring. Experimental results demonstrate that Multi-TCA can improve predictive performance on previously unseen domains.

1 Introduction

In many real-world applications one would like to make use of the knowledge acquired from related domains on previously unseen domains. This problem is known as *domain generalization*, and recently has started to gain attention in the machine learning community [12, 3]. *Domain adaptation* [14] and domain generalization are subareas of transfer learning, aiming to find a shared subspace for related domains. While domain adaptation methods require at least some input data from the target domains, domain generalization methods are designed to generalize to previously unseen domains.

Most machine learning techniques rely on the assumption that the entire data, both training and testing, underlies the same data generation process. However, this assumption is often violated when data originates from multiple domains. Inequalities in the data generation process can lead to significant differences in marginal and conditional distributions of the data. Traditional machine learning methods can handle these differences only in two non-optimal ways: *i)* an individual model is fitted for each domain; a large amount of data is required, which is expensive and the fitted models often do not generalize to new domains. *ii)* Differences in the data generation process are ignored, by learning a model on

the pooled data. This approach usually results in low prediction accuracy and poor generalization [15].

Considerable effort has been made to remedy this problem (see [15, 10] and references therein). Given one or more target domains, the idea of domain adaptation is to adapt a model trained on the training domains such that the generalization error on the test domains is minimized. The dissimilarities of data distributions from different domains are considered explicitly. Compared to a single model fit to all domains, predictive accuracy and the generalization to new domains can be improved. In comparison to tackling each domain independently, data can be used much more efficiently, as knowledge is transferred between domains. In this way, the effort to allocate data is massively reduced. The main drawback with this approach is that one has to re-train the models for every new target domain, which is time-consuming and inhibits real-time applications. Domain generalization is a solution to this problem: across-domain information is extracted from training data and can be used on the target domains without re-training.

The assumption in domain generalization is that the training and test domains are related. That is, there is at least some common information among the domains that is relevant for the considered machine learning task. Feature subsets of domain datasets can differ by a combination of various properties, including mean shift, scale, skewness, kurtosis, and rotation. For some applications such properties become obvious when doing exploratory data analysis. However, usually the description of the task and some domain knowledge already give strong indications to reject the hypothesis that the whole data is sampled from the same data generation process. For example, in medical applications, the data collected from two different patients usually cannot be assumed to be sampled from the same data generation process. See Section 2.2 and Section 4.1 for further examples.

Transfer Component Analysis (TCA) [14] is a domain adaptation technique that aims to learn a shared subspace between a source domain and a target domain. The shared subspace consists of some transfer components learned in a *reproducing kernel Hilbert space (RKHS)* [13] using *maximum mean discrepancy (MMD)* [5]. In the subspace spanned by these transfer components, data distributions of different domains should be close to each other and the task-relevant information of the original data should be preserved.

In this paper, we extend the formulation of TCA to multiple domains, compare it to the domain generalization method *Domain-Invariant Component Analysis (DICA)* Muandet *et al.*, and evaluate their benefits on real datasets. The same extension presented in this paper enables the use of TCA for *i)* domain adaptation problems with multiple domains; *ii)* domain generalization problems with multiple source and target domains. This paper focuses on the domain generalization setting. Our solution is based on the idea of learning a shared subspace between source domains and using this subspace for related target domains – without re-training. We present and evaluate two variants of our extension, an unsupervised version to which we refer as *Multiple-Domain Transfer*

Component Analysis (Multi-TCA) and a semi-supervised version called *Multiple-Domain Semi-Supervised Transfer Component Analysis (Multi-SSTCA)*.

The remainder of this paper is organized as follows: Section 2 discusses related work in domain generalization and the bigger area of transfer learning and domain adaptation. Section 3 presents our proposed extension to TCA in both supervised and unsupervised settings. Section 4 shows an experimental evaluation on two publicly available datasets from the UCI repository. Section 5 gives the conclusions and directions for future work.

2 Related Work

2.1 Domain Generalization

Although, there is a large amount of publications in the field of transfer learning and domain adaptation, very few studies address domain generalization. Recently, Muandet *et al.* [12] presented a method called *Domain-Invariant Component Analysis (DICA)*, which addresses the problem of domain generalization. DICA and its unsupervised version UDICA are closely related to Multi-SSTCA and Multi-TCA. UDICA and Multi-TCA are derived differently but have similar objectives. They both try to find a subspace where: *i)* the distance between the domain datasets is minimized; *ii)* the variance in the feature space is maximized. DICA is an extension of UDICA that takes the functional relationship between X and Y into account – the derivation is again different to the extension of Multi-TCA to Multi-SSTCA. Besides the different derivation, Multi-SSTCA is more versatile than DICA as *i)* Multi-SSTCA can also consider the *manifold information* (see objective 3 in Section 3.2); *ii)* the definition of Multi-SSTCA can handle missing class labels, allowing the application of Multi-SSTCA to semi-supervised domain generalization and domain adaptation tasks.

Persello and Bruzzone [17] address domain generalization by selecting features that minimize the shift in the domain dataset distributions. Their selection criteria select variables that have *i)* high dependency with the target variable and *ii)* invariant data distributions across domains.

2.2 Transfer Learning and Domain Adaptation

In contrast to domain generalization, transfer learning and domain adaptation have received a lot of attention in the recent years. Transfer learning [15] aims at transferring knowledge from some previous tasks to a target task when the latter has limited training data. Domain adaptation [18, 2, 10] can be viewed as a subdomain of transfer learning, that deals primarily with a mismatch between training and test input distributions. A popular and intuitive approach for domain adaptation is to make the source and target distributions as similar as possible. The methods that follow this line of research can be grouped into two categories. Firstly, sample re-weighting [8, 4] approaches, which apply weights to the source samples to adjust their influence in the source distribution. Secondly, learning a shared subspace is a common approach in settings

where there is distribution mismatch. A typical approach in multi-task learning is to uncover a latent feature space that is shared across tasks. Commonly, latent factors are designed to represent statistical properties and/or the geometric structure of the data. Methods of this category exist for problems with different feature spaces [21] or marginal distributions [11].

There are quite some successful applications of transfer learning methods in different real-world applications. One application is a WiFi-based indoor localization problem presented in [16]. In this application the data is highly dependent on time, space and the client device. Transfer learning was successfully applied to transfer localization models over these dependencies. Another application area is in the field of image processing, e.g. Hinton *et al.* [7] apply transfer learning in a face recognition and a handwritten digit example. In [20, 12], Varnek *et al.* apply transfer learning techniques in biological applications.

3 Transfer Component Analysis for Domain Generalization

TCA aims to learn a good feature representation across different distributions, i.e., a shared subspace. In the learned subspace the distance of the individual dataset distributions is minimized and properties of the data are preserved. The use of a RKHS provides the possibility to use non-linear kernels. Subsequently, any machine learning method for regression, classification or clustering can be used on the identified subspace.

TCA has originally been designed to work with the most common transfer learning setting. In this setting, the goal is to find a common representation for one source domain $\mathcal{D}_S$ and one target domain $\mathcal{D}_T$, with at least some input data X_S, X_T existing in both domains. Here, a kernel-induced feature map ϕ is learned from $\{X_S, X_T\}$. Once transformed, the combined source and target data can be used in the subsequent machine learning task. The TCA algorithm and the learning setting described in this paper are different to the original Paper presented by Pan *et. al* [14] in the following aspects:

- **Differences in the learning algorithm:** This paper gives an extension of TCA to more than two domains. This can simply be achieved by extending the cost, weight and kernel matrices – see Equation 1 for an extension of the cost function as well as Equation 2 and Equation 3 for extension of the matrices.
- **Differences in the learning task:** The original paper considers two domains with input data from both domains. However, in our application, the TCA transformation is applied to domains without any input data. Here, first a common subspace for the source domain datasets $X_1, ..., X_S$ is learned. The learned model can then be applied to the target domain datasets $X_{S+1}, ..., X_U$. The assumption is that the common data properties extracted from the source datasets also apply to the target data.

The reminder of this paper describes the extensions from TCA/SSTCA to Multi-TCA/Multi-SSTCA. See Pan *et al.* [14], and references within, for a more detailed description of TCA/SSTCA, especially for the derivation of formulas.

3.1 (Unsupervised) Transfer Component Analysis

Multi-TCA is applicable if $P(X_s) \neq P(X_u), 1 \leq s < u \leq U$, where X_s, X_u are domain datasets, $P(X_s)$ is the probability distribution of X_s and U is the total number of source and target domain datasets. The goal of Multi-TCA is to find a feature map ϕ such that $P(\phi(X_s)) \approx P(\phi(X_u))$.

Assume ϕ is a feature map induced by a universal kernel. *Maximum mean discrepancy (MMD)* [5] measures the distance between the empirical means of two domains in the RKHS. We extend to more than two domains

$$\text{MMD} = \frac{1}{S}\sum_{s=1}^{S} ||\mu_{x_s} - \mu_{\bar{x}}||^2_{\mathcal{H}}. \tag{1}$$

Here, $\mu_{x_s} = \frac{1}{n_s}\sum_{i=1}^{n_s}\phi(x_{si})$ and $\mu_{\bar{x}} = \frac{1}{S}\sum_{s=1}^{S}\mu_{x_s}$, where n_s are the number of instances from X_s. S is the number of training domain datasets, x_{s_i} denotes the i^{th} instance of X_s and $||\cdot||_{\mathcal{H}}$ is the RKHS norm.

Let K be a combined Gram matrix [19] of the cross-domain data of the training domain $X_1, X_2, ..., X_S$:

$$K = \begin{bmatrix} K_{X_1,X_1} & K_{X_1,X_2} & \dots & K_{X_1,X_S} \\ K_{X_2,X_1} & K_{X_2,X_2} & \dots & K_{X_2,X_S} \\ \vdots & \vdots & \ddots & \vdots \\ K_{X_S,X_1} & K_{X_S,X_2} & \dots & K_{X_S,X_S} \end{bmatrix} \in \mathbb{R}^{N\times N} \tag{2}$$

where $N = \sum_{s=1}^{S} n_s$. Each element $K_{i,j}$ of K is given by $\phi(x_i)^T\phi(x_j)$. The calculation of MMD in Equation 1 can be rewritten as $tr(KL)$, where $L_{i,j}$ is defined as

$$L_{i,j} = \begin{cases} \frac{S-1}{N^2 n_s^2} & \text{if } x_i, x_j \in X_s \\ -\frac{1}{N^2 n_s n_u} & \text{if } x_i \in X_s, x_j \in X_u \text{ and } s \neq u \end{cases} \tag{3}$$

and $s, u \in \{1, ..., S\}$. The computational expensive semi-definite programming can be avoided by the use of a parametric kernel map $K = (KK^{-1/2})(K^{-1/2}K)$. Pan *et. al* [14] shows that the resulting kernel matrix $\tilde{K} = KWW^TK$, where $W \in \mathbb{R}^{N\times m}$, $m \ll N$ is an orthogonal transformation matrix that is found by Multi-TCA. As a result the MMD distance in Equation 1 can be rewritten as

$$\text{MMD} = tr((KWW^TK)L) = tr(W^TKLKW). \tag{4}$$

Similarly to PCA and KPCA [19], the second objective of Multi-TCA is to maximally preserve the data variance. The variance of the projected samples is W^TKHKW, where centering matrix H is defined as $H = I - \frac{1}{N}\mathbf{1}\mathbf{1}^T$. Here,

$\mathbf{1} \in \mathbb{R}^N$ is a column vector with all ones and $I \in \mathbb{R}^{N \times N}$ is the identity matrix. With a regulation term $tr(W^T W)$ and the tradeoff parameter μ, the objective of Multi-TCA can be formulated as

$$\min_W tr(W^T KLKW) + \mu\ tr(W^T W),\ \text{s.t. } W^T KHKW = I. \tag{5}$$

The embedding of the data in the latent space is given by $W^T K$. The solution of W is given by the $m \ll N$ leading eigenvectors of

$$(KLK + \mu I)^{-1} KHK, \tag{6}$$

where $\mu > 0$ is a tradeoff parameter that is usually needed to control the complexity of W.

3.2 Semi-Supervised Transfer Component Analysis

Multi-SSTCA is an extension to Multi-TCA based on SSTCA from Pan *et al.* [14] that also considers the conditional probabilities $P(Y_i|X_i)$, $i \in 1, ..., S$ and optimizes the following three objectives:

1. *Distribution Matching:* as in Multi-TCA, the first objective is to minimize the distribution differences – measured by the MMD criterion (Equation 1) – between the domain datasets.
2. *Label Dependence:* maximize the dependency between the embedding and the labels. This is achieved by the use of the Hilbert-Schmidt Independence Criterion (HSIC) [6] given by $\max_{K \succeq 0} tr(HKHK_{yy})$, where $K_{yy} = \gamma_w K_l + (1 - \gamma_w) K_v$. Here, $k_l = \phi(y_i, y_j)$, $K_v = I$ and γ_w is a tradeoff parameter that balances the label dependence with the data variance terms. The second objective is to
$$\max_W tr(W^T KHK_{yy} HKW). \tag{7}$$
3. *Locality Preserving:* Multi-SSTCA uses the manifold regularization of Belkin *et al.* [1]. In order to preserve locality, each x_i and x_j that are neighbors in the input space should also be neighbors in the data's embedding. A matrix $M \in \mathbb{R}^{N \times N}$ is constructed by $M_{i,j} = \exp(-(x_i - x_j)^2 / 2\sigma^2)$ if x_i is one of the k nearest neighbors of x_j, and $M_{i,j} = 0$ otherwise. The graph Laplacian is defined by $A = D - M$, where $D \in \mathbb{R}^{N \times N}$ is a diagonal matrix with entries $D_{i,i} = \sum_{j=1}^{N} M_{i,j}$. The third objective is to
$$\min_W \sum_{(i,j) \in N} M_{i,j} ||[W^T K]_i - [W^T K]_j||^2 = tr(W^T KAKW). \tag{8}$$

For Multi-SSTCA, the objective function is

$$\begin{aligned} \min_W\ & tr(W^T KLKW) + \frac{\lambda}{n^2} tr(W^T KAKW) + \mu\ tr(W^T W) \\ & \text{s.t. } W^T KHK_{yy} HKW = I \end{aligned} \tag{9}$$

and the solution of W is given by the $m < N$ leading eigenvectors of

$$(K(L + \lambda A)K + \mu I)^{-1} KHK_{yy}HK. \tag{10}$$

Note that Multi-SSTCA is a semi-supervised method in the domain adaptation setting, where the input data from the target domain is used without the target data. In domain generalization problems no target data is used at all. Thus, in domain generalization the special case occurs where Multi-SSTCA is used in a supervised setting – provided that no labels are missing in the source domains. Despite of this technicality, the same method can be used for both problem settings, domain adaptation and domain generalization.

4 Experimental Evaluation

4.1 Experimental Setup

We use two datasets for the experimental evaluation. *i)* The landmine data represents a landmine detection problem [22], based on airborne synthetic-aperture radar measurements. It has 9 features and 29 domains. As the class labels (1 for landmine and 0 for clutter) are highly unbalanced, we took all instances with class 1 and randomly selected the same amount of class 0 examples in each repetition, resulting in a total number of 1808 instances.*ii)* The Parkinson telemonitoring dataset [9], which consists of biomedical voice measurements from 42 people with early-stage Parkinson's disease. The Parkinson data was collected in a six-month trial of a telemonitoring device for remote symptom progression monitoring (5875 recordings in total). The goal is to predict the clinician's scoring of Parkinson's disease symptom based on 16 voice measurements. There are two scores to predict: the motor score and the total score on the Unified Parkinson's Disease Rating Scale (UPDRS). We consider each dataset, related to one patient, as a domain.

We compared Multi-TCA and Multi-SSTCA as preprocessor for a linear SVM with: *i)* KPCA with an RBF kernel as preprocessor for a linear SVM; *ii)* an SVM with a linear kernel and an SVM with an RBF kernel without any preprocessing. For the landmine data 5 training domains are selected from each of *relatively highly foliated* (domains $1 - 15$) and *bare earth or desert* (domains $16 - 29$) regions. For the Parkinson data we also consider 10 training domains. For both datasets, the remaining domains are used for testing. We randomly repeat 25 times the selection of training and testing domains. Parameters are selected by 5-folds cross-validation.

On both datasets, the number of components for all preprocessors is selected from $\{1..15\}$. For the input data we use an RBF kernel and select $\gamma \in \{0.005..0.5\}$ and $\gamma \in \{0.1..1\}$ for the Parkinson data and the landmine data, respectively. For classification with DICA and Multi-SSTCA we apply the output kernel $k_{yy}(y_i, y_j) = 1$ if $y_i = y_j$ and -1 otherwise. For regression we use an RBF kernel with $\gamma = 0.1$. We set the Multi-TCA/Multi-SSTCA parameter $\mu = 0.1$ and the UDICA/DICA parameter $\lambda = 0.1$ for the landmine data. For the Parkinson data

$\mu = 0.01$ and $\lambda = 0.01$. For UDICA and DICA $\epsilon = 0.0001$. The Multi-SSTCA parameter $\gamma_w = 0.5$. For Multi-SSTCA we build one model considering the manifold information ($\lambda = 1000$) and one without considering manifold information ($\lambda = 0$). We construct A using an RBF kernel ($\gamma = 1$) and 4-nearest neighbors. For SVM we select $C \in \{10^{-4}..10^{4}\}$ and for γ we apply the same ranges that are used by the preprocessors.

4.2 Experimental Results

The relative performance of the algorithms are summarized in Table 1. Performance on the test data is measured by misclassification rate (MC) for the landmine data and root mean square error (RMSE) for the Parkinson data.

The results in Table 1 show that Multi-TCA perform best on the landmine data, followed by UDICA. DICA performs best on the Motor score Parkinson problems, closely followed by the performance of Multi-SSTCA. With the same RMSE of 8.73, Multi-SSTCA and DICA are also the best methods on the Total score Parkinson problem. While taking the labels into account is clearly beneficial on the Parkinson data, it is not on the landmine data, where Multi-SSTCA and DICA perform worse than their unsupervised versions and KPCA. For Multi-SSTCA and the evaluated datasets, the modeling of the manifold information does not lead to any considerable improvements.

Table 1. Predictive performances *mean*(*std*) of the evaluated methods.

Preprocessor	SVM Kernel	Parkinson Motor score	Parkinson Total score	Landmine
Multi-TCA	linear	**32.39 ±1.49**	11.39 ±0.76	8.91 ±0.69
Multi-SSTCA($\lambda = 0$)	linear	32.81 ±1.32	11.30 ±0.83	**8.73 ±0.77**
Multi-SSTCA($\lambda = 1000$)	linear	33.14 ±1.61	11.29 ±0.82	8.76 ±0.76
UDICA	linear	32.43 ±1.26	11.58 ±0.80	9.02 ±0.68
DICA	linear	33.56 ±1.20	**11.25 ±0.82**	**8.73 ±0.75**
KPCA	linear	32.71 ±1.53	11.53 ±0.85	8.89 ±0.70
None	linear	32.66 ±1.43	12.30 ±3.90	9.15 ±1.27
None	RBF	32.51 ±1.19	11.56 ±1.28	9.02 ±0.98

On the evaluated datasets, differences in computation time can be neglected for the domain generalization methods. For example, the required computation time of the considered methods (Multi-TCA, Multi-SSTCA, UDICA, DICA) on the Parkinson Motor Score data are all within 32 ± 1 seconds on a standard PC. The most computational demanding tasks are the eigenvalue decomposition followed by the computation of kernel matrix K (Equation 2), which are required by each of the methods.

5 Conclusion and Future Work

In this paper we presented an extension of TCA to multiple domains and successfully applied it for domain generalization. We showed that improvements in

predictive performance can be achieved by aligning related datasets via the domain generalization methods Multi-(SS)TCA and (U)DICA. The performances of Multi-TCA and Multi-SSTCA on the two benchmark datasets were comparable to the performances of UDICA and DICA, respectively. However, compared to DICA, Multi-SSTCA can also take the manifold information into account (locality preserving) and is applicable for semi-supervised domain generalization tasks and domain adaptation. Although, the manifold information did not lead to any considerable improvement on the evaluated datasets, Pan *et al.* [14] showed its usefulness on simulated and real world domain adaptation problems. In the future we want to investigate the usefulness of the manifold information for domain generalization problems. Multi-TCA/Multi-SSTCA has many parameters that can be optimized. We plan to conduct sensitivity analysis and work on a parameter selection strategy.

Acknowledgment Parts of this work were carried out in mpcEnergy, a project supported within the program Regionale Wettbewerbsfähigkeit OÖ 2007-2013 by the European Fund for Regional Development as well as the State of Upper Austria.

References

[1] Belkin, M.: Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. J. Mach. Learn. Res. 7, 2399–2434 (2006)
[2] Bickel, S., Brückner, M., Scheffer, T.: Discriminative learning under covariate shift. J. Mach. Learn. Res. 10, 2137–2155 (2009)
[3] Blanchard, G., Lee, G., Scott, C.: Generalizing from several related classification tasks to a new unlabeled sample. In: NIPS. pp. 2178–2186 (2011)
[4] Gong, B., Grauman, K., Sha, F.: Connecting the dots with landmarks: Discriminatively learning domain-invariant features for unsupervised domain adaptation. In: ICML. pp. 222–230 (2013)
[5] Gretton, A., Borgwardt, K., Rasch, M., Schölkopf, B., Smola, A.: A kernel method for the two-sample-problem. In: NIPS. pp. 513–520 (2006)
[6] Gretton, A., Bousquet, O., Smola, A., Schölkopf, B.: Measuring statistical dependence with Hilbert-Schmidt norms. In: ALT. pp. 63–77 (2005)
[7] Hinton, G.E., Salakhutdinov, R.: Using deep belief nets to learn covariance kernels for gaussian processes. In: NIPS. pp. 1249–1256 (2007)
[8] Huang, J., Smola, A.J., Gretton, A., Borgwardt, K.M., Schölkopf, B.: Correcting sample selection bias by unlabeled data. In: NIPS. pp. 601–608 (2006)
[9] Little, M., McSharry, P., Roberts, S., Costello, D., Moroz, I.: Exploiting nonlinear recurrence and fractal scaling properties for voice disorder detection. BioMedical Engineering OnLine 6:23(1) (2007)
[10] Long, M., Pan, S.J., St Yu, P., Wang, J., Ding, G.: Adaptation regularization: A general framework for transfer learning. IEEE Trans. on Know. and Data Eng. 26(5), 1076–1089 (2014)

[11] Long, M., Wang, J., Ding, G., Shen, D., Yang, Q.: Transfer learning with graph co-regularization. In: Proc. of the 26th Conf. on Art. Int. pp. 1805–1818. AAAI (2012)
[12] Muandet, K., Balduzzi, D., Schölkopf, B.: Domain generalization via invariant feature representation. In: Proc. of the 30th Int. Conf. on Mach. Learn. pp. 10–18 (2013)
[13] Müller, K., Mika, S., Rätsch, G., Tsuda, K., Schölkopf, B.: An introduction to kernel-based learning algorithms. IEEE Trans. on Neural Networks 12(2), 181–201 (2001)
[14] Pan, S.J., Tsang, I., Kwok, J., Yang, Q.: Domain adaptation via transfer component analysis. Trans. on Neural Networks 22(2), 199–210 (2011)
[15] Pan, S.J., Yang, Q.: A survey on transfer learning. IEEE Trans. on Know. and Data Eng. 22(10), 1345–1359 (2010)
[16] Pan, S.J., Zheng, V.W., Yang, Q., Hu, D.H.: Transfer learning for wifi-based indoor localization. In Proc. of the Workshop on Trans. Learn. for Complex Tasks, of the 23rd AAAI Conf. on Art. Int. pp. 43–48 (2008)
[17] Persello, C., Bruzzone, L.: Relevant and invariant feature selection of hyperspectral images for domain generalization. In: Int. Geoscience and Remote Sensing Symposium (IGARSS). pp. 3562–3565. IEEE (2014)
[18] Quionero-Candela, J., Sugiyama, M., Schwaighofer, A., Lawrence, N.D.: Dataset Shift in Machine Learning. The MIT Press (2009)
[19] Schölkopf, B., Smola, A., Müller, K.: Kernel principal component analysis (1999)
[20] Varnek, A., Gaudin, C., Marcou, G., Baskin, I., Pandey, A.K., Tetko, I.V.: Inductive transfer of knowledge: application of multi-task learning and feature net approaches to model tissue-air partition coefficients. J. of Chem. Inf. and Modeling 49(1), 133–144 (2009)
[21] Wang, C., Mahadevan, S.: Heterogeneous domain adaptation using manifold alignment. In: Proc. of the Twenty-Second int. joint conf. on Art. Int. vol. 2, pp. 1541–1546. AAAI (2011)
[22] Xue, Y., Liao, X., Carin, L., Krishnapuram, B.: Multitask learning for classication with Dirichlet process priors. J. Mach. Learn. Res. 35(8), 35–63 (2007)