Neural Process Lett (2017) 46:845–855
DOI 10.1007/s11063-017-9612-8

Multi-Domain Transfer Component Analysis for Domain Generalization

Thomas Grubinger[1] · **Adriana Birlutiu**[2] · **Holger Schöner**[1] · **Thomas Natschläger**[1] · **Tom Heskes**[3]

Published online: 6 April 2017

Abstract This paper presents the domain generalization methods Multi-Domain Transfer Component Analysis (Multi-TCA) and Multi-Domain Semi-Supervised Transfer Component Analysis (Multi-SSTCA) which are extensions of the domain adaptation method Transfer Component Analysis to multiple domains. Multi-TCA learns a shared subspace by minimizing the dissimilarities across domains, while maximally preserving the data variance. The proposed methods are compared to other state-of-the-art methods on three public datasets and on a real-world case study on climate control in residential buildings. Experimental results demonstrate that Multi-TCA and Multi-SSTCA can improve predictive performance on previously unseen domains. We perform sensitivity analysis on model parameters and evaluate different kernel distances, which facilitate further improvements in predictive performance.

The research reported in this paper has been supported by the Austrian Ministry for Transport, Innovation and Technology, the Federal Ministry of Science, Research and Economy, and the Province of Upper Austria in the frame of the COMET center SCCH.

✉ Thomas Grubinger
thomas.grubinger@scch.at

Adriana Birlutiu
adriana.birlutiu@uab.ro

Holger Schöner
holger.schoener@scch.at

Thomas Natschläger
thomas.natschlaeger@scch.at

Tom Heskes
t.heskes@science.ru.nl

1 Data Analysis Systems, Software Competence Center Hagenberg, Hagenberg im Mühlkreis, Austria

2 Faculty of Science, "1 Decembrie 1918" University of Alba-Iulia, Alba Iulia, Romania

3 Institute for Computing and Information Sciences, Radboud University Nijmegen, Nijmegen, The Netherlands

Keywords Domain generalization · Domain adaptation · Transfer learning · Transfer component analysis

1 Introduction

Transfer learning has emerged as a new machine learning framework that investigates how to recognize and apply knowledge and skills learned in previous tasks to novel tasks in new domains [14]. Standard supervised 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. Transfer learning algorithms extract knowledge from the source domain and transfer it to the target domain in order to improve the prediction function and/or speed up the learning of the target domain.

Domain adaptation [19,24] considers the dissimilarities of different domains explicitly, which allows the joint modeling of multiple domain datasets. In comparison to tackling each domain independently, data can be used much more efficiently, as knowledge is transferred between domains. The main drawback of this approach is that one has to collect data and re-train the models for every new target domain, which is time-consuming and inhibits real-time applications.

Domain generalization [2,11] is a solution to this problem: across-domain information is extracted from the source domain data (where training data is available) and can be used on the target domains (where no training data is available) without re-training. The assumption in domain generalization is that the source and target domains are related. *Transfer Component Analysis (TCA)* [13] is a popular domain adaptation technique that aims to learn a shared subspace between different domains. In the shared subspace, the data distributions of different domains should be close to each other and task-relevant information of the original data be preserved.

We present two extensions of TCA for domain generalization: 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)*. Multi-TCA and Multi-SSTCA are suited for domain generalization problems and domain adaptation problems with multiple source and target domains. We compare our methods to the recently proposed domain generalization method *Domain-Invariant Component Analysis (DICA)* [11].

This paper is an extension of [8]. The newly added contributions are: (i) a sensitivity analysis on the Multi-TCA, Multi-SSTCA, UDICA and DICA model parameters, (ii) the evaluation of alternative kernel distances ($\mathcal{X}^2$, Jensen–Shannon—see Sect. 4.1.1), which facilitate further improvements in predictive performance, (iii) extended evaluation on one additional public dataset (Graft-versus-Host Disease (GvHD) [3]) and (iv) a real world case study on climate control in residential buildings (3 additional datasets).

2 Related Work

Blanchard et al. [2] were among the firsts to approach domain generalization and proposed an SVM that encodes empirical marginal distributions.

Muandet et al. [11] introduced a feature-projection based algorithm, named *Domain-Invariant Component Analysis (DICA)*. DICA and its unsupervised version UDICA are closely related to the algorithms we propose in this paper. UDICA and Multi-TCA are derived in a different manner but have similar objectives. To find a subspace where: (i) the distance between the domain datasets is minimized, (ii) the variance in the feature space is maximized. 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 Sect. 3.2), (iii) 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 [15] address domain generalization by selecting features that minimize the shift in the domain dataset distributions. Domain generalization algorithms have also been used for object recognition tasks [4,21].

An intuitive approach for domain adaptation is to make the source and target distributions as similar as possible. This can be achieved by sample re-weighting [5] approaches that apply weights to the source samples to adjust their influence in the target distribution. Feature representation approaches learning a shared subspace are very well suited in settings where there is a distribution mismatch [17]. An alternative approach for multiple source domains is to combine a weighted combination of predictors that are learned on single domains [18,22].

3 Transfer Component Analysis for Domain Generalization

TCA aims to learn a good feature representation across different distributions. The use of a *reproducing kernel Hilbert space (RKHS)* [12] 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 for domain adaptation with two domains.

The TCA algorithm and the learning setting described in this paper are different from the original paper presented by Pan et. al [13] in the following aspects: **(i) Differences in the learning algorithm:** This paper gives an extension of TCA to more than two domains. This can be achieved by extending the cost, weight and kernel matrices—see Eq. 1 for an extension of the cost function and Eqs. 2 and 3 for extensions of the matrices. **(ii) 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 learned from multiple source domain datasets $X_1, \ldots, X_S$. The learned model can then be applied to the target domain datasets $X_{S+1}, \ldots, X_U$. 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. [13] for a more detailed description of TCA/SSTCA, especially for the derivation of formulas.

3.1 Multi-Domain 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)* [6] 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 source 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 [16] of the cross-domain data of the source domain $X_1, X_2, \ldots, X_S$:

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

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 Eq. 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 $s, u \in \{1, \ldots, S\}$. The computationally expensive semi-definite programming can be avoided by a parametric kernel map $K = (KK^{-1/2})(K^{-1/2}K)$. Pan et. al [13] 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 TCA. As a result the MMD distance in Eq. 1 can be rewritten as $\text{MMD} = tr((KWW^TK)L) = tr(W^TKLKW)$.

Similarly to PCA and KPCA [16], the second objective of Multi-TCA is to maximally preserve the data variance. The variance of the projected samples is W^TKHKW, where the 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 $tr(W^TW)$ and the tradeoff parameter μ, the objective of Multi-TCA can be formulated as $\min_W tr(W^TKLKW) + \mu\, tr(W^TW)$, s.t. $W^TKHKW = I$.

The embedding of the data in the latent space is given by W^TK. The solution of W is given by the $m \ll N$ leading eigenvectors of $(KLK + \mu I)^{-1}KHK$, where $\mu > 0$ is a tradeoff parameter that is usually needed to control the complexity of W.

3.2 Multi-Domain Semi-Supervised Transfer Component Analysis

Multi-SSTCA is an extension to Multi-TCA based on SSTCA from Pan et al. [13] that also considers the conditional probabilities $P(Y_i|X_i)$, $i \in 1, \ldots, S$ and optimizes the following three objectives:

1. *Distribution Matching* as in Multi-TCA, the first objective is to minimize the distances (MMD in Eq. 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) [7] 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 $\max_W tr(W^TKHK_{yy}HKW)$.
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 $\min_W \sum_{(i,j)\in N} M_{i,j}||[W^T K]_i - [W^T K]_j||^2 = tr(W^T KAKW)$.

For Multi-SSTCA, the objective is $\min_W tr(W^T KLKW) + \frac{\lambda}{n^2} tr(W^T KAKW) + \mu\ tr(W^T W)$, s.t. $W^T KHK_{yy}HKW = I$ and the solution of W is given by the $m < N$ leading eigenvectors of $(K(L + \lambda A)K + \mu I)^{-1} KHK_{yy}HK$.

4 Experimental Evaluation

4.1 Evaluation of Three Publicly Available Datasets

We use three datasets in our experiments. (i) The Landmine data represents a landmine detection problem [23], 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 [10], which consists of biomedical voice measurements from 42 people with early-stage Parkinson's disease (5875 recordings in total). The goal is to predict the clinician's scoring (two objectives: motor score and the total score) of Parkinson's disease symptom based on 16 voice measurements. We consider each dataset, related to one patient, as a domain. (iii) Graft-versus-Host Disease (GvHD) data consists of weekly peripheral blood samples obtained from 31 patients following allogenic blood and marrow transplant. We use the data of the first 10 patients (10,000 recordings in total).

4.1.1 Experimental Setup

In 4.1 we apply an RBF kernel $K(x, y) = \exp(-\frac{1}{2\sigma^2 D^2(x,y)})$ with $D^2(x, y) = ||x - y||_2^2$ for all preprocessors. Replacing the L2 distance $||x - y||_2^2$ with another distance leads to the generalized RBF kernel. For the Parkinson and the GvHD data we apply the following commonly used distance measures (see. e.g., [20]): $\mathcal{X}^2$: $k(x, y) = 2(xy)/(x+y)$ and *Jensen–Shannon (JS)*: $k(x, y) = (x/2) \times \log_2(x + y)/x + (y/2) \times \log_2(x + y)/y$. Other possible choices for the kernel function include: Hellinger, intersection and PQ [9]. Note that these alternative distance measures are only applicable if $X \in \mathcal{R}_+^{N\times m}$, where N and m are the number of instances and features, respectively—in the Parkinson telemonitoring and the climate control study in Sect. 4.2: $X \in \mathcal{R}^{N\times m}$.

We compared Multi-TCA and Multi-SSTCA as preprocessor for a linear SVM with: (i) the domain generalization methods UDICA and DICA with an RBF kernel as preprocessor for a linear SVM, (ii) KPCA with an RBF kernel as preprocessor for a linear SVM, (iii) an SVM with a linear kernel, an SVM with an RBF kernel and *k-nearest neighbor (KNN)* without any preprocessing. For the Landmine data 5 source 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 and the GvHD data we consider 10 and 3 domains for training. For all datasets, the remaining domains are used for testing. We randomly repeat 25 times the selection of source and target domains. Parameters are selected by fivefolds cross-validation.

The number of components for all preprocessors is selected from $1, 2, \ldots 15$ for the Parkinson telemonitoring and Landmine detections datasets, and from 25, 35, 50, 75 for the GvHD data. For the input data we use an RBF kernel and select $\gamma \in \{0.005, 0.01, 0.025, 0.05, \ldots, 0.5\}$ for Landmine and $\gamma \in \{0.1, 0.25, 0.5, 1\}$ for Parkinson and GvHD. 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 Landmine and GvHD. For Parkinson $\mu = 0.01$ 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 ($\lambda = 0$). We construct A using an RBF kernel ($\gamma = 1$) and 4-nearest neighbors. For SVM we select $C \in \{10^{-4}, 10^{-3} \ldots 10^{4}\}$ and for γ we apply the same ranges that are used by the preprocessors.

4.1.2 Experimental Results

The results in Table 1a show that Multi-TCA perform best on Landmine, followed by UDICA. DICA performs best on the Motor score Parkinson problem, closely followed by Multi-SSTCA. With the same RMSE of 8.73, Multi-SSTCA and DICA are also the best methods on the Total score Parkinson problem. On GvHD, Multi-TCA performs best with 8.03 followed by UDICA (8.16). While taking the conditional probabilities into account is clearly beneficial on the Parkinson data, it is not for Landmine and GvHD, where Multi-SSTCA and DICA perform worse than their unsupervised versions.

Table 1b, shows that the Jensen–Shannon (JS) and $\mathcal{X}^2$ distance compared to the L2 distance. Small improvements are achieved on Parkinson (11.25 for DICA with L2 distance compared to 11.23 for KPCA with $\mathcal{X}^2$ distance on the Motor score problem and 8.73 for Multi-SSTCA and DICA with L2 distance to 8.71 for DICA with JS distance). A much larger improvement is archived for GvHD, where the best performance of 8.03, for Multi-TCA with L2 distance can be improved to 6.77 for Multi-TCA and UDICA.

4.2 Climate Control in Residential Buildings

Three transfer learning problems for climate control (heating/cooling) in residential buildings are evaluated. The domains are given by data from different residential buildings, which mainly differ by size, presence pattern and weather conditions (collected from different locations).

Let u_t, y_t denote the tth sample of control input data U and target variable Y. The input data for y_t can be written as $y_{t-1}, \ldots, y_{t-\ell}, u_{t-1}, \ldots, u_{t-\ell}$. Here, u_t includes all variables directly or indirectly affecting the indoor temperature, such as the flow of the thermal medium, the inlet/outlet temperatures of the thermal medium, the occupant's presence, the outdoor temperature and the solar gain. The target variable y in our evaluation is the room temperature of the main living In our experiment $\ell = 3$ and the control interval $t = 0.5$ h.

Table 2 depicts important properties of the three learning problems. Weather data was collected from Linz, Austria, (*source house*) and Washington, DC, (*target houses*) from December to February 2009. Two different presence patterns were used. Presence pattern 2 differs from presence pattern 1 by the number of people and their activities (type and frequency). A house with *size* 3 indicates a floor space three times larger than a house with *size* 1.

Table 1 Performances ($mean(std)$) of the evaluated methods on the 3 public datasets

Preprocessor Method	Kernel Dist.	Landmine	Parkinson Motor score	Parkinson Total score	GvHD
(a) Results with standard RBF Kernel (L2 distance) for all preprocessors [The best result for every dataset is marked in bold]					
Multi-TCA	L2	**32.39 (1.49)**	11.39 (0.76)	8.91 (0.69)	**8.03 (1.19)**
Multi-SSTCA ($\lambda = 0$)	L2	32.81 (1.32)	11.30 (0.83)	**8.73 (0.77)**	9.17 (1.28)
Multi-SSTCA ($\lambda = 1000$)	L2	33.14 (1.61)	11.29 (0.82)	8.76 (0.76)	10.19 (2.42)
UDICA	L2	32.43 (1.26)	11.58 (0.80)	9.02 (0.68)	8.16 (1.08)
DICA	L2	33.56 (1.20)	**11.25 (0.82)**	**8.73 (0.75)**	8.29 (1.10)
KPCA	L2	32.71 (1.53)	11.53 (0.85)	8.89 (0.70)	9.57 (0.94)
SVM (linear)	–	32.66 (1.43)	12.30 (3.90)	9.15 (1.27)	9.93 (1.27)
SVM (RBF)	–	32.51 (1.19)	11.56 (1.28)	9.02 (0.98)	8.35 (1.13)
KNN	–	35.44 (1.47)	12.67 (0.64)	9.60 (0.46)	8.25 (0.98)
(b) Results with alternative kernel distances [marked if there is an improvement over the best result in (a)]					
Multi-TCA	$\mathcal{X}^2$	Not app.	11.31 (0.83)	8.74 (0.72)	**6.77 (1.10)**
Multi-TCA	JS	Not app.	11.30 (0.79)	8.82 (0.71)	7.22 (0.99)
Multi-SSTCA ($\lambda = 0$)	$\mathcal{X}^2$	Not app.	11.30 (0.85)	8.76 (0.72)	19.50 (6.48)
Multi-SSTCA ($\lambda = 0$)	JS	Not app.	11.29 (0.86)	8.78 (0.68)	11.08 (6.29)
Multi-SSTCA ($\lambda = 1000$)	$\mathcal{X}^2$	Not app.	11.28 (0.78)	8.76 (0.74)	12.55 (7.64)
Multi-SSTCA ($\lambda = 1000$)	JS	Not app.	11.33 (0.83)	8.78 (0.68)	7.45 (1.07)
UDICA	$\mathcal{X}^2$	Not app.	11.30 (0.89)	8.76 (0.72)	**6.77 (1.01)**
UDICA	JS	Not app.	11.31 (0.88)	8.77 (0.75)	7.53 (0.97)
DICA	$\mathcal{X}^2$	Not app.	11.35 (0.90)	8.81 (0.75)	9.51 (3.74)
DICA	JS	Not app.	11.27 (0.84)	**8.71 (0.76)**	7.95 (3.26)
KPCA	$\mathcal{X}^2$	Not app.	**11.23 (0.82)**	8.73 (0.74)	23.54 (0.21)
KPCA	JS	Not app.	11.27 (0.80)	8.74 (0.72)	8.41 (1.41)

Misclassification rate (MC) is used for the Landmine and GvHD and root mean square error (RMSE) for Parkinson

Table 2 Description of the 6 source and 3 target scenarios

Scen	Exp. 1	Exp. 2	Exp. 3	Usage	Weather Loc.	Size	Presence
S1	Yes	No	Yes	Source scenario	Linz, Austria	1	Pattern 1
S2							Pattern 2
S3	No	Yes	Yes			2	Pattern 1
S4							Pattern 2
S5	Yes	Yes	No			3	Pattern 1
S6							Pattern 2
T1	No	Yes	No	Target scenario	Washington, DC	1	Pattern 1
T2	Yes	No	No			2	Pattern 1
T3	No	No	Yes			3	Pattern 1

Yes/No in columns *Exp. 1–Exp. 3* indicate in which experiments the respective scenarios are used. Column *Usage* indicates whether the scenario is used as source or target

Table 3 Performances (root mean square error (RMSE), reported as $mean(std)$) of the experiments depicted in Table 2

Preprocessor	Kernel distance	Exp. 1	Exp. 2	Exp. 3
Multi-TCA	L2	0.156 (0.005)	**0.181 (0.005)**	**0.151 (0.005)**
Multi-SSTCA ($\lambda = 0$)	L2	0.292 (0.068)	0.242 (0.034)	0.256 (0.027)
Multi-SSTCA ($\lambda = 1000$)	L2	0.281 (0.058)	0.231 (0.024)	0.243 (0.044)
UDICA	L2	**0.155 (0.006)**	**0.181 (0.005)**	**0.151 (0.006)**
DICA	L2	0.458 (0.657)	0.319 (0.648)	0.324 (0.467)
PCA	L2	0.159 (0.006)	0.193 (0.011)	0.158 (0.008)
SVM (linear)	–	0.159 (0.005)	0.194 (0.010)	0.161 (0.011)
SVM (RBF)	–	0.174 (0.011)	0.198 (0.011)	0.177 (0.023)
KNN	–	0.991 (0.027)	0.198 (0.011)	0.954 (0.051)

4.2.1 Experimental Setup

Similar to Sect. 4.1, we compare Multi-TCA and Multi-SSTCA with UDICA, DICA and PCA as preprocessors for a linear SVM. In contrast to the previous section, we use a linear kernel for all preprocessors and linear PCA instead of KPCA—in case of the climate control in residential building data, a linear kernel leads to much better results for all preprocessing methods. We again compare to a linear SVM, a SVM with an RBF kernel and KNN without any preprocessing. We randomly select 250 datapoints from each source domain and use all target scenario data (≈4400) for testing.

On all three problems the number of components for all preprocessors is selected from 5,10,20,30. For the Multi-TCA/Multi-SSTCA parameter μ is selected from $1, 10 \dots 1000$ and the UDICA/DICA parameter $\lambda \in \{1, 10 \dots 1000\}$. 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 ($\gamma = 0.1$) and 4-nearest neighbors. For SVM we select $C \in \{10^{-4}, 10^{-3} \dots 10^{4}\}$ and for $\gamma \in \{10^{-4}, 10^{-3} \dots 10^{-1}\}$.

4.2.2 Experimental Results

The performances on the 3 climate control in residential building problems are summarized in Table 3. The results show that Multi-TCA and UDICA perform best on all 3 experiments—with nearly equivalent performance on all 3 experiments—followed by PCA, which performs slightly worse. Note that the differences between Multi-TCA and PCA are very small, but consistent and with low standard deviation. For example, in Exp. 1, Multi-TCA has an RMSE of 0.156 and PCA of 0.159, but Multi-TCA performs better on 20 out of the 25 evaluations. Taking the conditional probabilities into account is clearly not beneficial in this application. Multi-SSTCA and DICA give unreliable prediction—especially the DICA performance estimates have very high standard deviations on all three experiments. For Multi-SSTCA, the modeling of the manifold information (Multi-SSTCA with $\lambda = 1000$) leads to slightly better results than Multi-SSTCA with $\lambda = 0$.

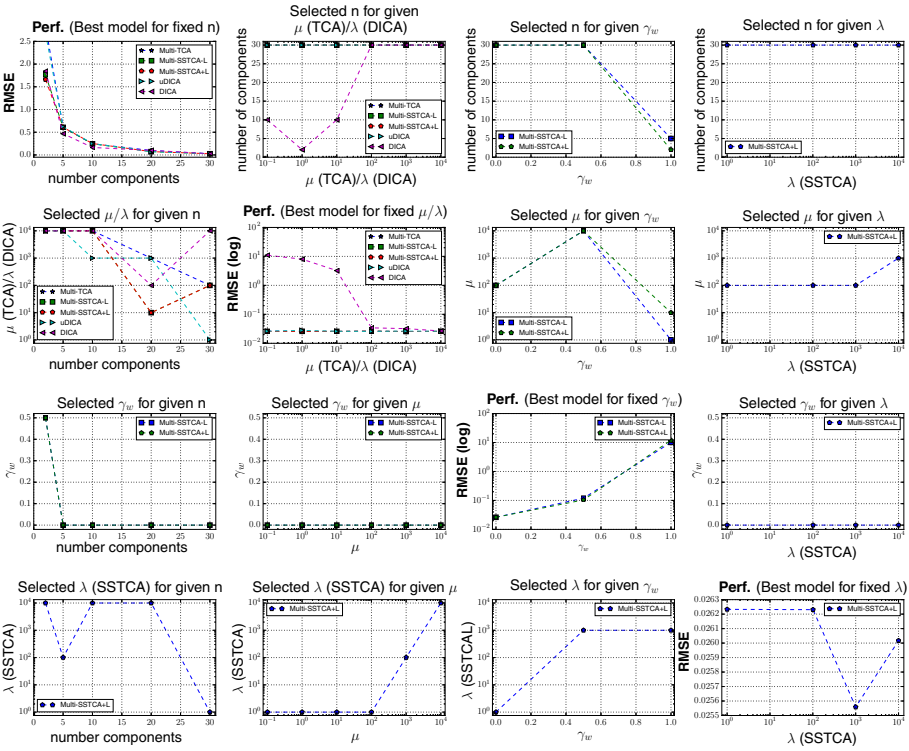

Fig. 1 Sensitivity analysis on the Multi-TCA/Multi-SSTCA and UDICA/DICA parameters on Exp. 1 of the climate control in residential building data

4.2.3 Sensitivity Analysis of Model Parameters

In this section we conduct a sensitivity analysis on the most important parameters of Multi-TCA/Multi-SSTCA and UDICA/DICA. For all four methods we vary the number of components from $2, 3 \ldots 30$. We investigate the Multi-TCA/Multi-SSTCA parameter μ and the UDICA/DICA parameter λ in the range of $10^{-1}, 10^{0} \ldots 10^{4}$. Note that, as μ and λ in UDICA/DICA have a similar effect, we visualize them in the same plots. For Multi-SSTCA we include two versions: (i) Multi-SSTCA-L, here we vary $\gamma_w \in \{0, 0.5, 1\}$, but leave out the manifold information (Multi-SSTCA parameter $\lambda = 0$). (ii) Multi-SSTCA+L, where we investigate the effect of the Multi-SSTCA parameter $\lambda \in \{0, 10^1, 10^2, 10^3, 10^4\}$. For each of the three experiments, we deterministically select 250 equidistant data points from each source scenario. For testing, we again select the whole data from the respective target scenario.

The analysis is conducted by fixing specific parameter values for one parameter and conducting a parameter search over the remaining parameters—on the source scenarios. The upper left graph in Fig. 1 shows the performance—RMSE on the target scenario—for different fixed number of components. The three subplots below indicate which parameters were selected for each model, fitted with the predefined number of components. In the same way the second column shows the influence of fixed μ (Multi-TCA/Multi-SSTCA) values and fixed λ (UDICA/DICA) values. The third column continues with fixing Multi-SSTCA

parameter γ_w and the fourth column indicates the influence of Multi-SSTCA parameter λ. Note that the influence of the fixed parameter on the RMSE is always plotted on the diagonal, while the remaining parameters are always plotted below and above the fixed parameter.

Let $sp[i, j]$ indicate the subplot in the i-th column and j-th row. If not stated otherwise, the observations in this paragraph hold for all three experiments. $sp[1, 1]$ indicates that increasing numbers of components,—within our considered range from $\{2, 3 \ldots 30\}$—increase prediction performance. In $sp[2, 2]$, we see that large values for λ (UDICA/DICA) are clearly the best choice for DICA, while the parameter has not much influence on UDICA's performance. Also the choice of μ for (Multi-TCA/Multi-SSTCA) does not seem to have a big effect on the methods' predictive accuracy. As illustrated in $sp[3, 3]$, a low γ_w is beneficial for both Multi-SSTCA versions. In particular, $\gamma_w = 0$ is the best value, in which case the Multi-SSTCA is purely unsupervised and does not consider the label information at all. $sp[4, 4]$ shows the, rather small, influence of λ (Multi-SSTCA+L) on the RMSE. The best values for λ is 10^2. $sp[4, 2]$ indicates that higher μ are selected for predefined choices of λ (Multi-SSTCA+L). $sp[4, 3]$ shows that $\gamma_w = 0$, indicating that only pure unsupervised models are selected.

5 Conclusions

In this paper we presented an extension of TCA to multiple domains and successfully applied it for domain generalization. We demonstrated 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 three benchmark datasets were comparable to the performances of UDICA and DICA, respectively. On the 3 problems from the climate control in residential building case study, Multi-TCA and UDICA performed best with nearly the same performance.

Compared to DICA, Multi-SSTCA can also take the manifold information into account and is applicable for semi-supervised domain generalization tasks and domain adaptation. On the climate control in residential building datasets, the Multi-SSTCA version using the manifold information (locality preserving) performed better than the Multi-SSTCA without the manifold information.

The sensitivity analysis on the climate control in residential building data showed important relations between the model parameters of Multi-TCA/Multi-SSTCA and UDICA/DICA to the predictive accuracy, as well as important interactions between the model parameters. Furthermore we showed, that predictive performance on the individual datasets can be further improved by the use of alternative distance measures, such as Jensen–Shannon or χ^2.

References

1. Belkin M (2006) Manifold regularization: a geometric framework for learning from labeled and unlabeled examples. J Mach Learn Res 7:2399–2434
2. Blanchard G, Lee G, Scott C (2011) Generalizing from several related classification tasks to a new unlabeled sample. In: NIPS, pp 2178–2186
3. Brinkman R, Gasparetto M, Lee SJ, Ribickas A, Perkins J, Janssen W, Smiley R, Smith C (2007) High-content flow cytometry and temporal data analysis for defining a cellular signature of graft-versus-host disease. Biol Blood Marrow Transplant 13(6):691–700
4. Ghifary M, Bastiaan Kleijn W, Zhang M, Balduzzi D (2015) Domain generalization for object recognition with multi-task autoencoders. In: Proceedings of the IEEE international conference on computer vision, pp 2551–2559

5. Gong B, Grauman K, Sha F (2013) Connecting the dots with landmarks: Discriminatively learning domain-invariant features for unsupervised domain adaptation. In: ICML, pp 222–230
6. Gretton A, Borgwardt K, Rasch M, Schölkopf B, Smola A (2006) A kernel method for the two-sample-problem. In: NIPS, pp 513–520
7. Gretton A, Bousquet O, Smola A, Schölkopf B (2005) Measuring statistical dependence with Hilbert-Schmidt norms. In: ALT, pp 63–77
8. Grubinger T, Birlutiu A, Schöner H, Natschläger T, Heskes T (2015) Domain generalization based on transfer component analysis. In: Advances in computational intelligence. Springer, pp 325–334
9. Ionescu RT, Popescu M (2015) PQ kernel: a rank correlation kernel for visual word histograms. Pattern Recognit Lett 55:51–57
10. Little M, McSharry P, Roberts S, Costello D, Moroz I (2007) Moroz I (2007) Exploiting nonlinear recurrence and fractal scaling properties for voice disorder detection. BioMed Eng OnLine 6(1):23
11. Muandet K, Balduzzi D, Schölkopf B (2013) Domain generalization via invariant feature representation. In: Proceedings of the 30th international conference on machhine learning, pp 10–18
12. Müller K, Mika S, Rätsch G, Tsuda K, Schölkopf B (2001) An introduction to kernel-based learning algorithms. IEEE Trans Neural Netw 12(2):181–201
13. Pan S, Tsang I, Kwok J, Yang Q (2011) Domain adaptation via transfer component analysis. IEEE Trans Neural Netw 22(2):199–210
14. Pan S, Yang Q (2010) A survey on transfer learning. IEEE Trans Knowl Data Eng 22(10):1345–1359
15. Persello C, Bruzzone L (2014) Relevant and invariant feature selection of hyperspectral images for domain generalization. In: International geoscience and remote sensing symposium (IGARSS), IEEE. pp 3562–3565
16. Schölkopf B, Smola A, Müller K (1999) Kernel principal component analysis. In: International Conference on Artificial Neural Networks, pp 583–588
17. Sun H, Liu S, Zhou S (2016) Discriminative subspace alignment for unsupervised visual domain adaptation. In: NEPL, pp 1–15
18. Sun S, Shi H (2013) Bayesian multi-source domain adaptation. In: International conference on machine learning and cybernetics, IEEE, vol 1, pp 24–28
19. Sun S, Shi H, Wu Y (2015) A survey of multi-source domain adaptation. Inf Fusion 24:84–92
20. Vedaldi A, Zisserman A (2012) Efficient additive kernels via explicit feature maps. IEEE Trans Pattern Anal Mach Intell 34(3):480–492
21. Xu Z, Li W, Niu L, Xu D (2014) Exploiting low-rank structure from latent domains for domain generalization. In: Computer vision—ECCV 2014—13th European conference, pp 628–643. doi:10.1007/978-3-319-10578-9_41
22. Xu Z, Sun S (2012) Multi-source transfer learning with multi-view adaboost. In: International conference on neural information processing systems, Springer. pp 332–339
23. Xue Y, Liao X, Carin L, Krishnapuram B (2007) Multitask learning for classication with Dirichlet process priors. J Mach Learn Res 35(8):35–63
24. Zhang H, Ji H, Wang X (2012) Transfer learning from unlabeled data via neural networks. NEPL 36(2):173–187