Domain Generalization via Invariant Feature Representation

Krikamol Muandet[1], **David Balduzzi**[2], and **Bernhard Schölkopf**[1]

[1]Empirical Inference Department, MPI for Intelligent Systems, Tübingen, Germany
[2]Machine Learning Laboratory, ETH Zurich, Zurich, Switzerland

Abstract

This paper investigates domain generalization: How to take knowledge acquired from an arbitrary number of related domains and apply it to previously unseen domains? We propose Domain-Invariant Component Analysis (DICA), a kernel-based optimization algorithm that learns an invariant transformation by minimizing the dissimilarity across domains, whilst preserving the functional relationship between input and output variables. A learning-theoretic analysis shows that reducing dissimilarity improves the expected generalization ability of classifiers on new domains, motivating the proposed algorithm. Experimental results on synthetic and real-world datasets demonstrate that DICA successfully learns invariant features and improves classifier performance in practice.

Domain Generalization

Standard Setting: Assume that the training data and test data come from the same distribution, learn a classifier/regressor that generalizes well to the test data.

Domain Adaptation: the training data and test data may come from different distributions. The common assumption is that we observe the test data at the training time. Adapt the classifier/regressor trained using the training data to the specific set of test data.

Covariate Shift: The marginal $\mathbb{P}(X)$ changes, but the conditional $\mathbb{P}(Y|X)$ stays the same.

Target Shift/Concept Drift The marginal $\mathbb{P}(Y)$ or conditional $\mathbb{P}(Y|X)$ may also change.

Domain Generalization: The training data comes from different distributions. Learn a classifier/regressor that generalizes well to the unseen test data, which also comes from different distribution.

Applications: medical diagnosis: aggregating the diagnosis of previous patients to the new patients who have similar demographic and medical profiles.

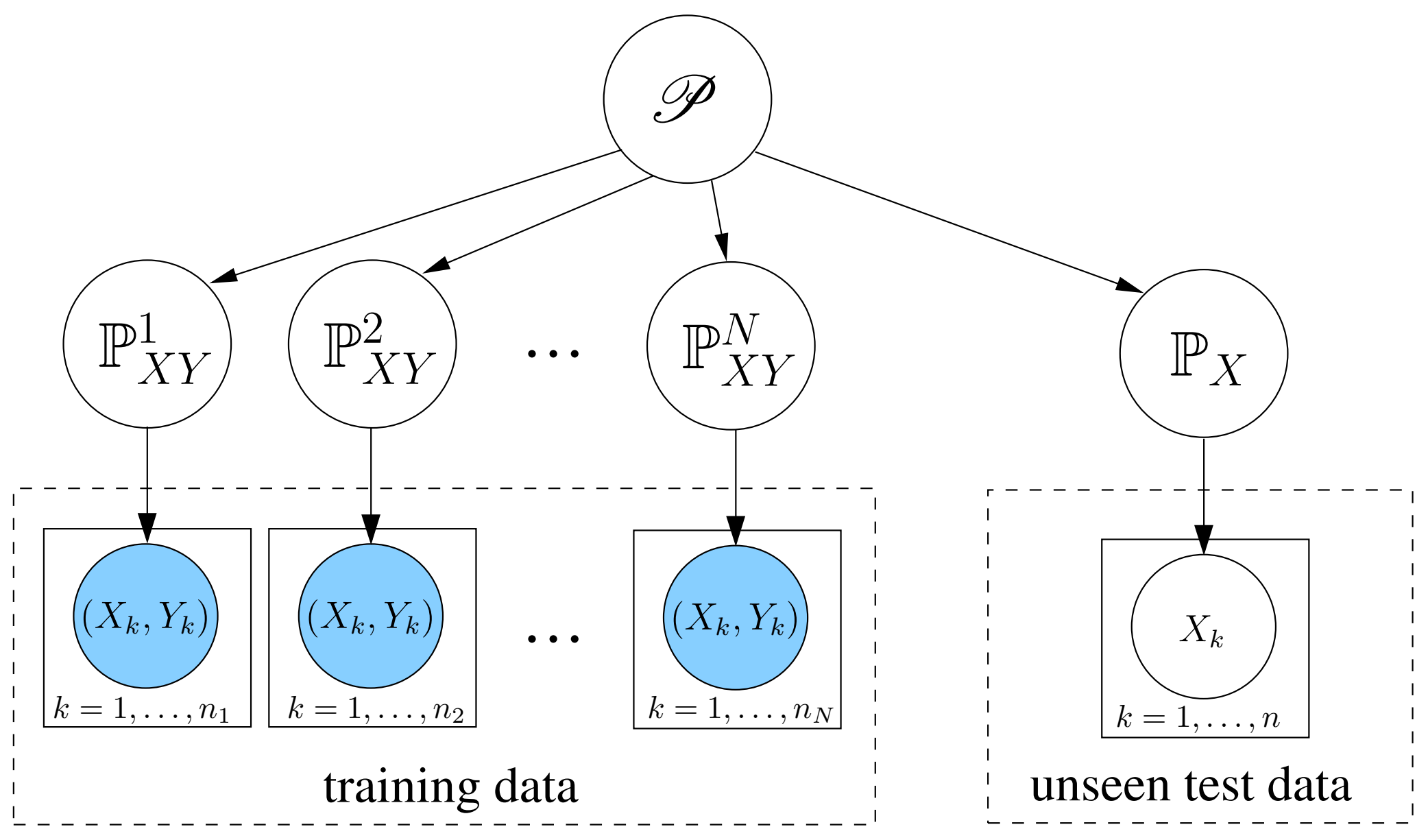

Figure 1: A simplified schematic diagram of the domain generalization framework. A major difference between our framework and most previous work in domain adaptation is that we do not observe the test domains during training time.

Objective

Domain 1	Domain 2		Domain N		New Domain
$\mathbb{P}^1_{XY} = \mathbb{P}^1_X\mathbb{P}^1_{Y\|X}$	$\mathbb{P}^2_{XY} = \mathbb{P}^2_X\mathbb{P}^2_{Y\|X}$	$\cdots$	$\mathbb{P}^N_{XY} = \mathbb{P}^N_X\mathbb{P}^N_{Y\|X}$	$\Longrightarrow$	$\mathbb{P}^t_X$
$S^1 = \{x^{(1)}_k, y^{(1)}_k\}^{n_1}_{k=1}$	$S^i = \{x^{(2)}_k, y^{(2)}_k\}^{n_2}_{k=1}$		$S^i = \{x^{(N)}_k, y^{(N)}_k\}^{n_N}_{k=1}$		$S^t = \{x^{(t)}_k\}^{n_t}_{k=1}$

Given the training sample $\mathcal{S}$, our goal is to produce an estimate $f : \mathfrak{P}_{\mathcal{X}} \times \mathcal{X} \to \mathbb{R}$ that generalizes well to test samples $S^t = \{x^{(t)}_k\}^{n_t}_{k=1}$. To actively reduce the dissimilarity between domains, we find transformation $\mathcal{B}$ in the RKHS $\mathcal{H}$ that

1. **minimizes the distance between empirical distributions of the transformed samples $\mathcal{B}(S^i)$ and**
2. **preserves the functional relationship between X and Y, i.e., $Y \perp X \,|\, \mathcal{B}(X)$.**

① Minimizing Distributional Variance

Distributional variance $\mathbb{V}_{\mathcal{H}}(\mathcal{P})$ estimates the variance of $\mathscr{P}_X$ which generates $\mathbb{P}^1_X, \mathbb{P}^2_X, \ldots, \mathbb{P}^N_X$.

Definition 1 *Introduce probability distribution $\mathcal{P}$ on $\mathcal{H}$ with $\mathcal{P}(\mu_{\mathbb{P}^i}) = \frac{1}{N}$ and center G to obtain the covariance operator of $\mathcal{P}$, denoted as $\Sigma := G - \mathbf{1}_N G - G\mathbf{1}_N + \mathbf{1}_N G \mathbf{1}_N$. The **distributional variance** is $\mathbb{V}_{\mathcal{H}}(\mathcal{P}) := \frac{1}{N}\mathrm{tr}(\Sigma) = \frac{1}{N}\mathrm{tr}(G) - \frac{1}{N^2}\sum_{i,j=1}^{N} G_{ij}$.*

The empirical distributional variance can be computed by

$$\widehat{\mathbb{V}}_{\mathcal{H}}(\mathcal{B}\mathcal{S}) = \mathrm{tr}(\widetilde{K}Q) = \mathrm{tr}(KBB^\top KQ) = \mathrm{tr}(B^\top KQKB)$$

② Preserving Functional Relationship

The central subspace C is the minimal subspace that captures the functional relationship between X and Y, i.e., $Y \perp X \,|\, C^\top X$.

Theorem 1 *If there exists a **central subspace** $C = [\mathbf{c}_1, \ldots, \mathbf{c}_m]$ satisfying $Y \perp X | C^\top X$, and for any $a \in \mathbb{R}^d$, $\mathbb{E}[a^\top X | C^\top X]$ is linear in $\{\mathbf{c}_i^\top X\}^m_{i=1}$, then $\mathbb{E}[X|Y] \subset$ span$\{\Sigma_{\mathrm{xx}}\mathbf{c}_i\}^m_{i=1}$.*

It follows that the bases C of the central subspace coincide with the m largest eigenvectors of $\mathbb{V}(\mathbb{E}[X|Y])$ premultiplied by $\Sigma^{-1}_{\mathrm{xx}}$. Thus, the basis $\mathbf{c}$ is the solution to the eigenvalue problem $\mathbb{V}(\mathbb{E}[X|Y])\Sigma_{\mathrm{xx}}\mathbf{c} = \gamma\Sigma_{\mathrm{xx}}\mathbf{c}$.

Domain-Invariant Component Analysis

Combining ① and ②, DICA finds $B = [\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \ldots, \boldsymbol{\beta}_m]$ that solves

$$\max_{B \in \mathbb{R}^{n \times m}} \frac{\frac{1}{n}\mathrm{tr}\left(B^\top L(L + n\varepsilon I_n)^{-1}K^2 B\right)}{\mathrm{tr}\left(B^\top KQKB + BKB\right)} \tag{1}$$

which leads to the following algorithms:

DICA Algorithm

```
Input:    Parameters $\lambda$, $\varepsilon$, and $m \ll n$.
          Sample $\mathcal{S} = \{S^i = \{(x^{(i)}_k, y^{(i)}_k)\}^{n_i}_{k=1}\}^N_{i=1}$.
Output:   Projection $B_{n\times m}$ and kernel $\widetilde{K}_{n\times n}$.
1: Calculate gram matrix $[K_{ij}]_{kl} = k(x^{(i)}_k, x^{(j)}_l)$ and $[L_{ij}]_{kl} = l(y^{(i)}_k, y^{(j)}_l)$.
2: Supervised: $C = L(L + n\varepsilon I)^{-1}K^2$.
3: Unsupervised: $C = K^2$.
4: Solve $\frac{1}{n}CB = (KQK + K + \lambda I)B\Gamma$ for $B$.
5: Output $B$ and $\widetilde{K} \leftarrow KBB^\top K$.
6: The test kernel $\widetilde{K}^t \leftarrow K^t BB^\top K$ where $K^t_{n_t\times n}$ is the joint kernel
   between test and training data.
```

A Learning-Theoretic Bound

Theorem 2 *Under reasonable technical assumptions, it holds with probability at least $1 - \delta$ that,*

$$\sup_{\|f\|_{\mathcal{H}} \le 1} \left| \mathbb{E}^*_{\mathscr{P}} \mathbb{E}_{\mathbb{P}} \ell(f(\tilde{X}_{ij}\mathcal{B}), Y_i) - \mathbb{E}_{\hat{\mathbb{P}}} \ell(f(\tilde{X}_{ij}\mathcal{B}), Y_i) \right|^2$$
$$\le c_1 \frac{1}{N}\mathrm{tr}(B^\intercal KQKB)$$
$$+\mathrm{tr}(B^\top KB)\left(c_2 \frac{N(\log\frac{1}{\delta} + 2\log N)}{n} + \frac{c_3 \log\frac{1}{\delta} + c_4}{N}\right).$$

The bound reveals a tradeoff between reducing the distributional variance and the complexity or size of the transform used to do so. The denominator of (1) is a sum of these terms, so that DICA tightens the bound in Theorem 2.

Preserving the functional relationship (i.e. central subspace) by maximizing the numerator in (1) should reduce the empirical risk $\mathbb{E}_{\hat{\mathbb{P}}}\ell(f(\tilde{X}_{ij}\mathcal{B}), Y_i)$, but a rigorous demonstration has yet to be found.

Relations to Existing Methods

The DICA and UDICA algorithms generalize many well-known dimension reduction techniques. In the supervised setting, if dataset $\mathcal{S}$ contains samples drawn from a single distribution $\mathbb{P}_{XY}$ then we have $KQK = \mathbf{0}$. Substituting $\alpha := KB$ gives the eigenvalue problem $\frac{1}{n}L(L + n\varepsilon I)^{-1}K\alpha = K\alpha\Gamma$, which corresponds to covariance operator inverse regression (COIR) [KP11].

If there is only a single distribution then unsupervised DICA reduces to KPCA since $KQK = \mathbf{0}$ and finding B requires solving the eigensystem $KB = B\Gamma$ which recovers KPCA [SSM98]. If there are two domains, source $\mathbb{P}_S$ and target $\mathbb{P}_T$, then UDICA is closely related – though not identical to – Transfer Component Analysis [Pan+11]. This follows from the observation that $\mathbb{V}_{\mathcal{H}}(\{\mathbb{P}_S, \mathbb{P}_T\}) = \|\mu_{\mathbb{P}_S} - \mu_{\mathbb{P}_T}\|^2$.

Experimental Results

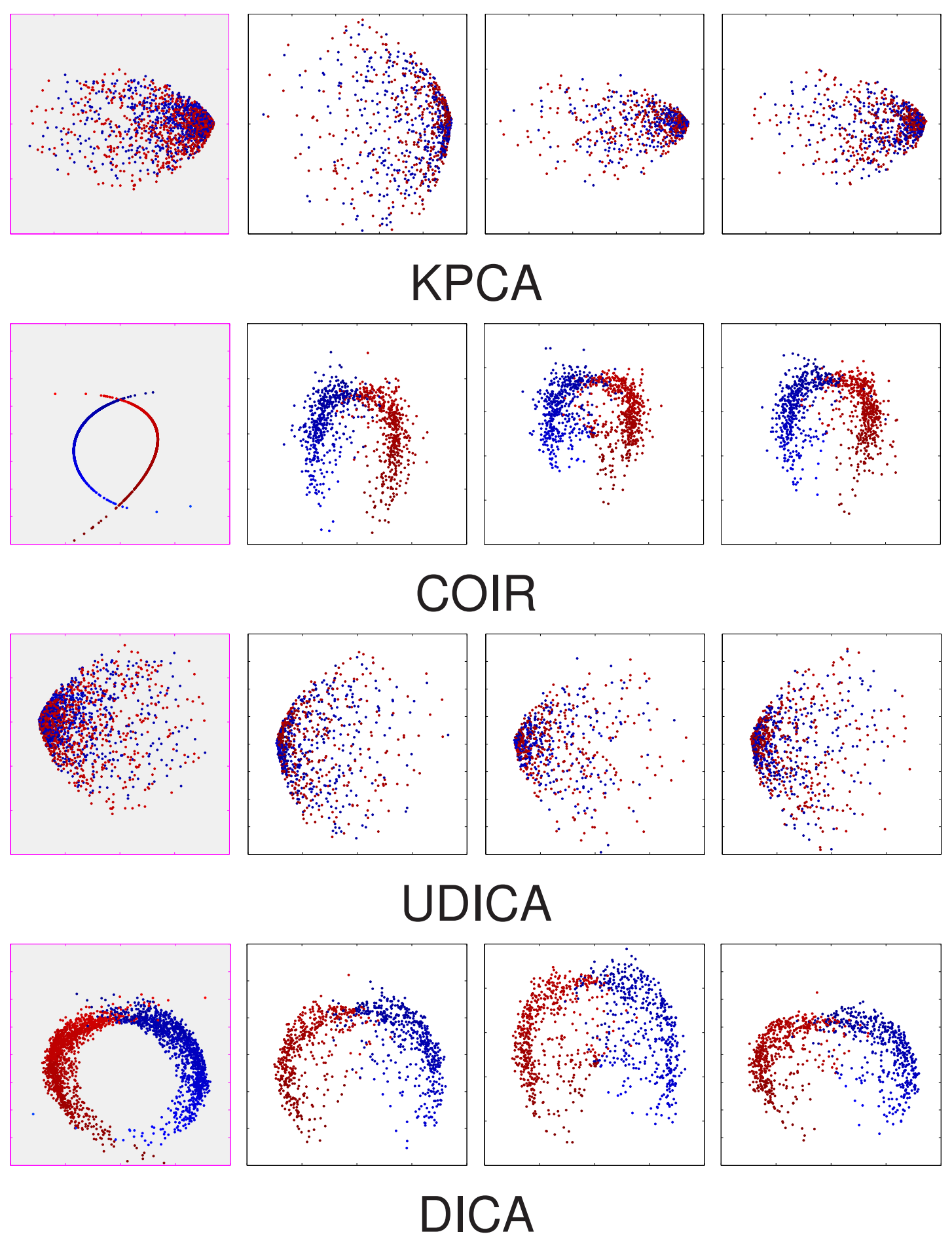

Figure 2: Projections of a synthetic dataset onto the first two eigenvectors obtained from the KPCA, UDICA, COIR, and DICA. The colors of data points corresponds to the output values. The shaded boxes depict the projection of training data, whereas the unshaded boxes show projections of unseen test datasets.

Pooling SVM applies standard kernel function on the pooled data from multiple domains.

Distributional SVM uses the kernel $K(\tilde{x}^{(i)}_k, \tilde{x}^{(j)}_l) = k_1(\mathbb{P}^i, \mathbb{P}^j) \cdot k_2(x^{(i)}_k, x^{(j)}_l)$.

Table 1: Average accuracies over 30 random subsamples of GvHD datasets. Pooling SVM applies standard kernel function on the pooled data from multiple domains, whereas distributional SVM also considers similarity between domains using kernel on distributions. With sufficiently many samples, DICA outperforms other methods in both pooling and distributional settings.

Methods	Pooling SVM			Distributional SVM		
	$n_i = 100$	$n_i = 500$	$n_i = 1000$	$n_i = 100$	$n_i = 500$	$n_i = 1000$
Input	91.68±.91	92.11±1.14	93.57±.77	91.53±.76	92.81±.93	92.41±.98
KPCA	91.65±.93	92.06±1.15	93.59±.77	**91.83±.60**	90.86±1.98	92.61±1.12
COIR	**91.71±.88**	92.00±1.05	92.57±.97	91.42±.95	91.54±1.14	92.61±.89
UDICA	91.20±.81	92.21±.19	93.02±.77	91.51±.79	91.74±1.08	93.02±.77
DICA	91.37±.91	**92.71±.82**	**94.16±.73**	91.51±.89	**93.42±.73**	**93.33±.86**

Table 2: The average leave-one-out accuracies over 30 subjects on GvHD data. The distributional SVM outperforms the pooling SVM. DICA improves classifier accuracy.

Methods	Pooling	Distributional
Input	92.03±8.21	93.19±7.20
KPCA	91.99±9.02	93.11±6.83
COIR	92.40±8.63	92.92±8.20
UDICA	92.51±5.09	92.74±5.01
DICA	**92.72±6.41**	**94.80±3.81**

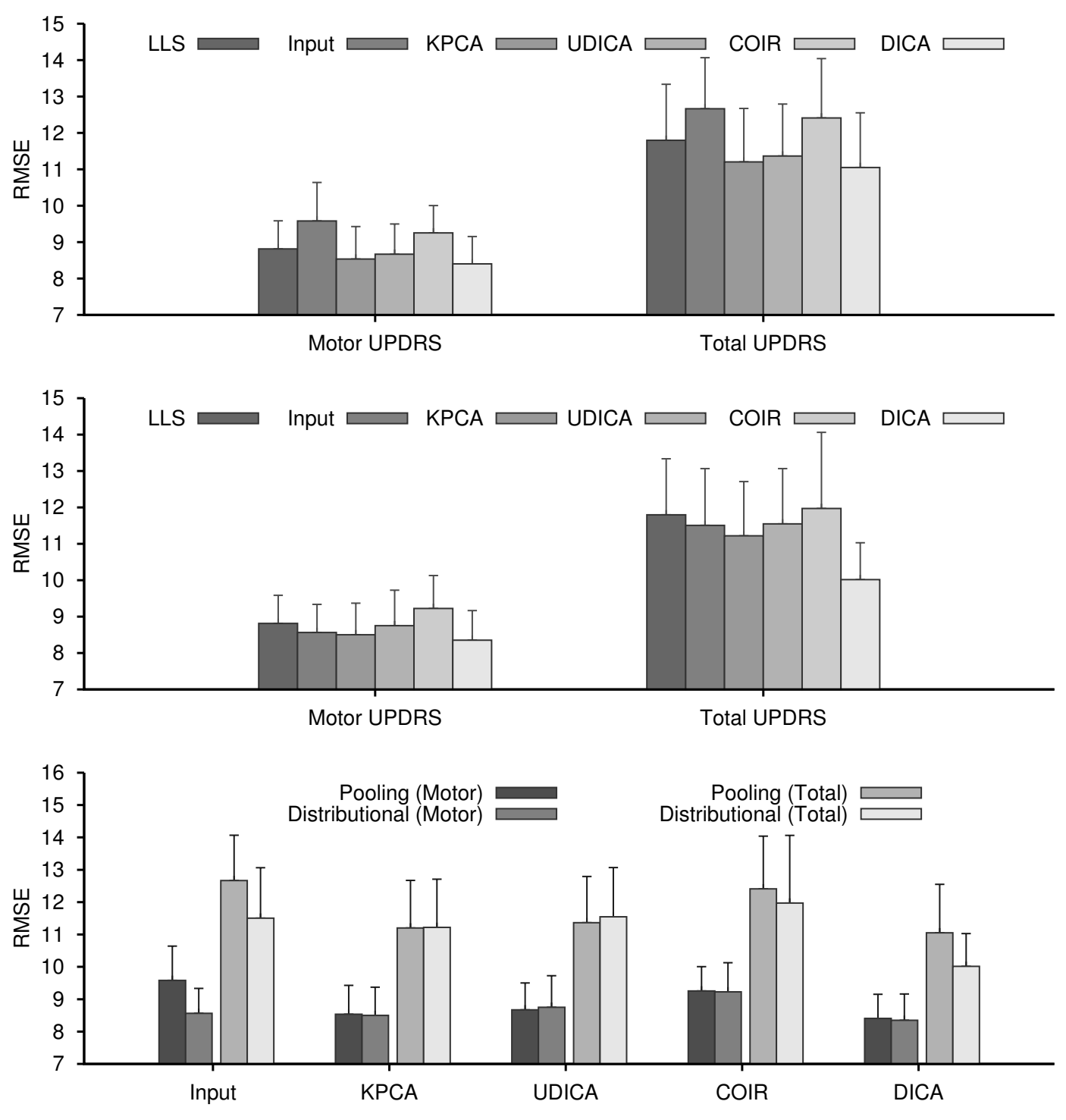

Figure 3: The root mean square error (RMSE) of motor and total UPDRS scores predicted by GP regression after different preprocessing methods on Parkinson's telemonitoring dataset. The top and middle rows depicts the pooling and distributional settings; the bottom row compares the two settings. Results of linear least square (LLS) are given as a baseline.

Conclusions

Domain-Invariant Component Analysis (DICA) is a new algorithm for domain generalization based on learning an invariant transformation of the data. The algorithm is theoretically justified and performs well in practice.

References

[KP11] M. Kim and V. Pavlovic. "Central subspace dimensionality reduction using covariance operators". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33.4 (2011), pp. 657–670.

[Pan+11] Sinno Jialin Pan et al. "Domain adaptation via transfer component analysis". In: *IEEE Transactions on Neural Networks* 22.2 (2011), pp. 199–210.

[SSM98] B. Schölkopf, A. Smola, and K-R. Müller. "Nonlinear component analysis as a kernel eigenvalue problem". In: *Neural Computation* 10.5 (July 1998), pp. 1299–1319.