Test-time Adaptation for Regression by Subspace Alignment

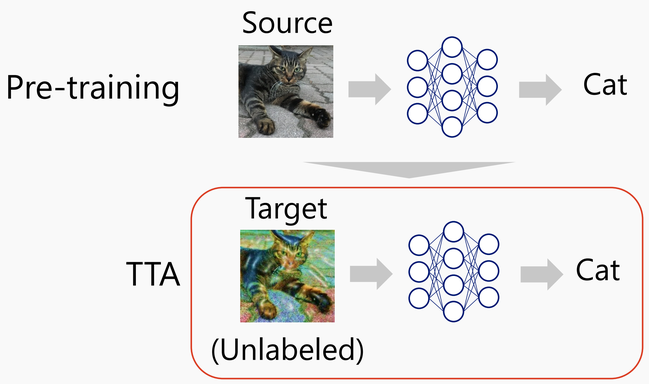
Kazuki Adachi[*‡], Shin'ya Yamaguchi[*†], Atsutoshi Kumagai[*], and Tomoki Hamagami[‡]
[*]NTT Corporation [†]Kyoto University [‡]Yokohama National University

Paper Code

NTT

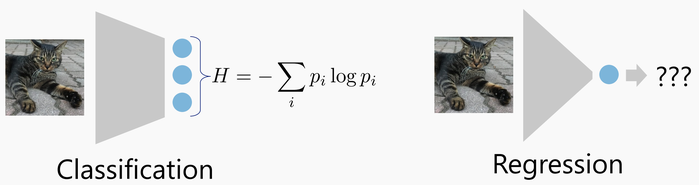
Test-time Adaptation (TTA)

- Adapt a pre-trained model to the target domain with unlabeled target data
- Not accessing the source data



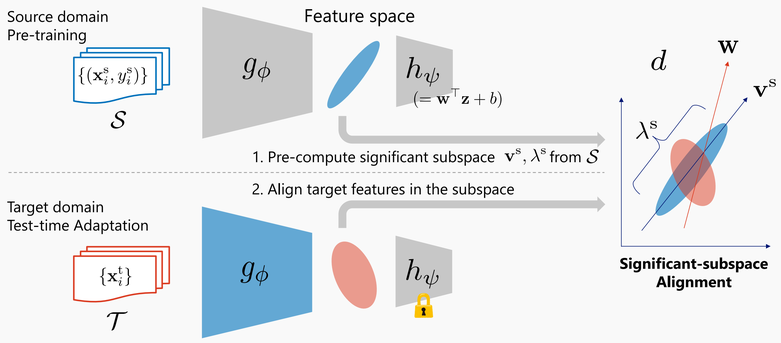
TTA for Regression

- TTA for regression has not been explored
- TTA methods for classification rely on entropy minimization
- Entropy cannot be computed for regression models (scalar outputs)
- TTA for regression not relying on entropy is necessary



Proposed Method

- Approach: Feature alignment in the subspace



- Basic idea: Feature alignment
  - Align the target feature mean and variance with pre-computed source statistics

$$\mathcal{L}_{\mathrm{TTA}}(\phi) = \sum_{d=1}^{D} D_{\mathrm{KL}}\left(\mathcal{N}(\mu_d^{\mathrm{s}}, \sigma_d^{\mathrm{s}\,2}) \| \mathcal{N}(\hat{\mu}_d^{\mathrm{t}}, \hat{\sigma}_d^{\mathrm{t}\,2})\right) + D_{\mathrm{KL}}\left(\mathcal{N}(\hat{\mu}_d^{\mathrm{t}}, \hat{\sigma}_d^{\mathrm{t}\,2}) \| \mathcal{N}(\mu_d^{\mathrm{s}}, \sigma_d^{\mathrm{s}\,2})\right)$$

  - **Problem**: Alignment in the entire feature space is inefficient
- Subspace detection
  - Features are distributed only in a small subspace (Tab. 1)
  - Detect the source feature subspace significant to the output using PCA
- Dimension weighting
  - Weight the subspace dimensions based on the significance to the output

$$\mathcal{L}_{\mathrm{TTA}}(\phi) = \sum_{d=1}^{K} \alpha_d \left\{ D_{\mathrm{KL}}\left(\mathcal{N}(0, \lambda_d^{\mathrm{s}}) \| \mathcal{N}(\tilde{\mu}_d^{\mathrm{t}}, \tilde{\sigma}_d^{\mathrm{t}\,2})\right) + D_{\mathrm{KL}}\left(\mathcal{N}(\tilde{\mu}_d^{\mathrm{t}}, \tilde{\sigma}_d^{\mathrm{t}\,2}) \| \mathcal{N}(0, \lambda_d^{\mathrm{s}})\right) \right\}$$

$$= \frac{1}{2} \sum_{d=1}^{K} \alpha_d \left(\frac{(\tilde{\mu}_d^{\mathrm{t}})^2 + \lambda_d^{\mathrm{s}}}{\tilde{\sigma}_d^{\mathrm{t}\,2}} + \frac{(\tilde{\mu}_d^{\mathrm{t}})^2 + \tilde{\sigma}_d^{\mathrm{t}\,2}}{\lambda_d^{\mathrm{s}}} - 2 \right) \qquad \alpha_d = 1 + |\mathbf{w}^\top \mathbf{v}_d^{\mathrm{s}}|$$

Experiment

- Subspace dimensions
  - Smaller than appearance (2048 dims.) in regression
  - Much smaller than in classification

Table 1. Number of feature dimensions

Dataset	Classification		Regression	
	#Valid dims	#Subspace dims.	#Valid dims	#Subspace dims.
SVHN	1946	64	353	14
CIFAR10	1521	86	561	50
UTKFace	2048	1471	2041	76
Biwi Kinect (mean)	2048	277	713	34.5
California Housing (100 dims.)	100	100	45	40

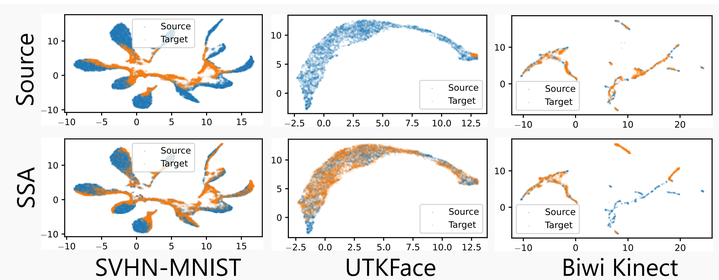
- Regression performance

Table 2. SVHN-MNIST

Method	R^2(↑)	RMSE (↓)	MAE (↓)
Source	0.406	2.232	1.608
DANN	$0.307_{\pm 0.09}$	$2.406_{\pm 0.16}$	$1.489_{\pm 0.09}$
TTT	$0.288_{\pm 0.02}$	$2.443_{\pm 0.03}$	$1.597_{\pm 0.03}$
BN-adapt	$0.396_{\pm 0.00}$	$2.251_{\pm 0.01}$	$1.458_{\pm 0.00}$
Prototype	$0.491_{\pm 0.00}$	$2.065_{\pm 0.01}$	$1.479_{\pm 0.01}$
FR	$0.369_{\pm 0.01}$	$2.300_{\pm 0.02}$	$1.631_{\pm 0.02}$
VM	$-685.1_{\pm 27.63}$	$75.83_{\pm 1.52}$	$75.78_{\pm 1.52}$
RSD	$0.252_{\pm 0.12}$	$2.497_{\pm 0.20}$	$1.703_{\pm 0.20}$
SSA (ours)	$\mathbf{0.511}_{\pm 0.03}$	$\mathbf{2.024}_{\pm 0.06}$	$\mathbf{1.209}_{\pm 0.04}$
Oracle	$0.874_{\pm 0.00}$	$1.028_{\pm 0.00}$	$0.575_{\pm 0.00}$

Table 3. UTKFace (age prediction, R^2)

Method	Defocus blur	Motion blur	Zoom blur	Contrast	Elastic transform	Jpeg comp.	Pixelate	Gaussian noise	Impulse noise	Shot noise	Brightness	Fog	Snow	Mean
Source	0.410	0.159	0.658	−3.906	0.711	0.069	0.595	−2.536	−2.539	−2.522	0.661	−0.029	−0.544	−0.678
DANN	0.512	0.586	0.637	−0.720	0.729	0.698	0.807	−4.341	−3.114	−3.744	0.590	−0.131	−0.425	−0.609
TTT	0.748	0.761	0.773	0.778	0.826	0.772	0.861	0.525	0.532	0.477	0.775	0.397	0.493	0.671
BN-Adapt	0.727	0.759	0.763	0.702	0.826	0.778	0.850	0.510	0.510	0.446	0.790	0.392	0.452	0.654
Prototype	−1.003	−1.020	−1.016	−0.719	−0.967	−0.908	−0.974	−0.514	−0.512	−0.512	−1.004	−0.823	−0.822	−0.830
FR	0.794	**0.839**	0.849	0.756	**0.899**	0.825	**0.946**	0.509	0.522	0.458	0.861	0.408	0.428	0.700
VM	−2.009	−1.991	−2.037	−1.889	−1.918	−1.918	−1.751	−2.181	−2.207	−2.176	−1.927	−2.250	−2.197	−2.035
RSD	0.789	0.833	0.851	0.749	0.897	0.825	0.941	0.502	0.503	0.445	0.862	0.419	0.500	0.701
SSA (ours)	**0.803**	**0.839**	**0.851**	**0.792**	**0.899**	**0.829**	0.943	**0.580**	**0.592**	**0.560**	**0.863**	**0.440**	**0.517**	**0.731**
Oracle	0.856	0.890	0.889	0.862	0.917	0.873	0.960	0.635	0.652	0.635	0.895	0.519	0.671	0.789



Feature visualization with UMAP