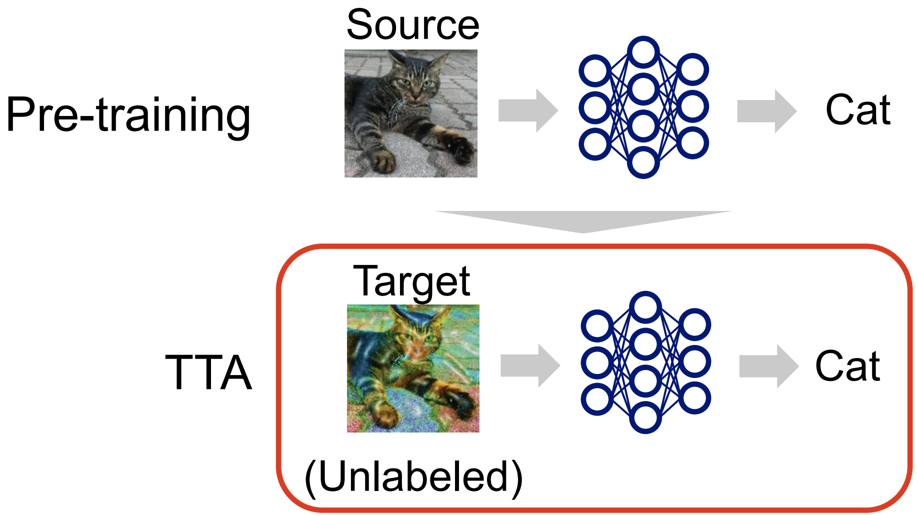
Test-time Adaptation for Regression by Subspace Alignment

Kazuki Adachi[*‡], Shin'ya Yamaguchi[*†], Atsutoshi Kumagai[*], Tomoki Hamagami[‡]

[*]NTT Computer and Data Science Laboratories
[†]Kyoto University
[‡]Yokohama National University

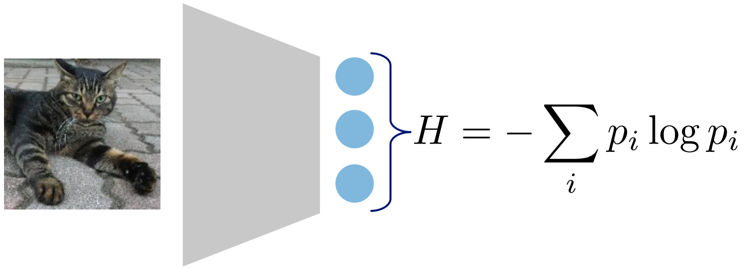
Test-time Adaptation (TTA)

- Adapts a pre-trained model to the target domain with unlabeled target data
- Does not access the source data

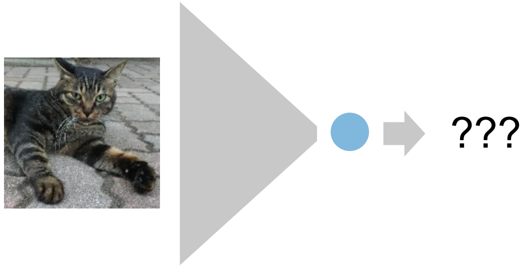


TTA for Regression

- TTA for regression has not been explored
- Existing TTA methods are typically designed for classification using entropy minimization
- Entropy cannot be computed for regression models!
  - Regression models output single scaler values, not distributions



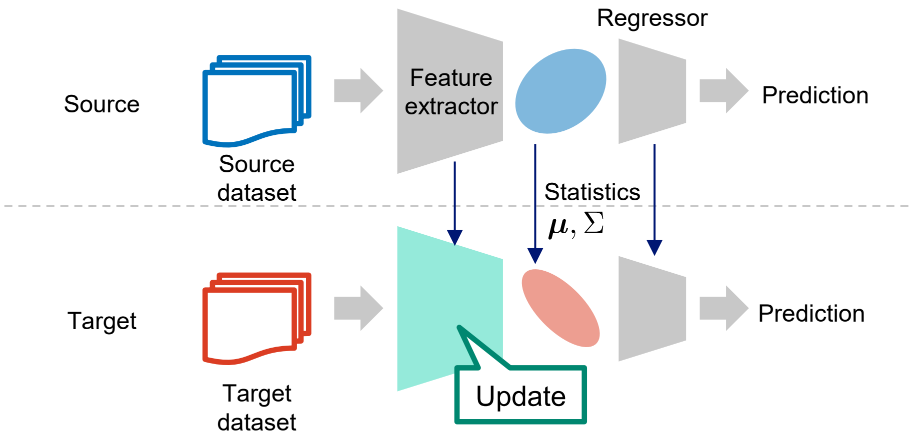
Classification



Regression

Proposed Method

- **Basic idea: Feature alignment**
  - Aligns the target feature statistics (mean and variance) with the pre-computed source ones
  - **Problem**: Alignment in the entire feature space is inefficient



Proposed Method

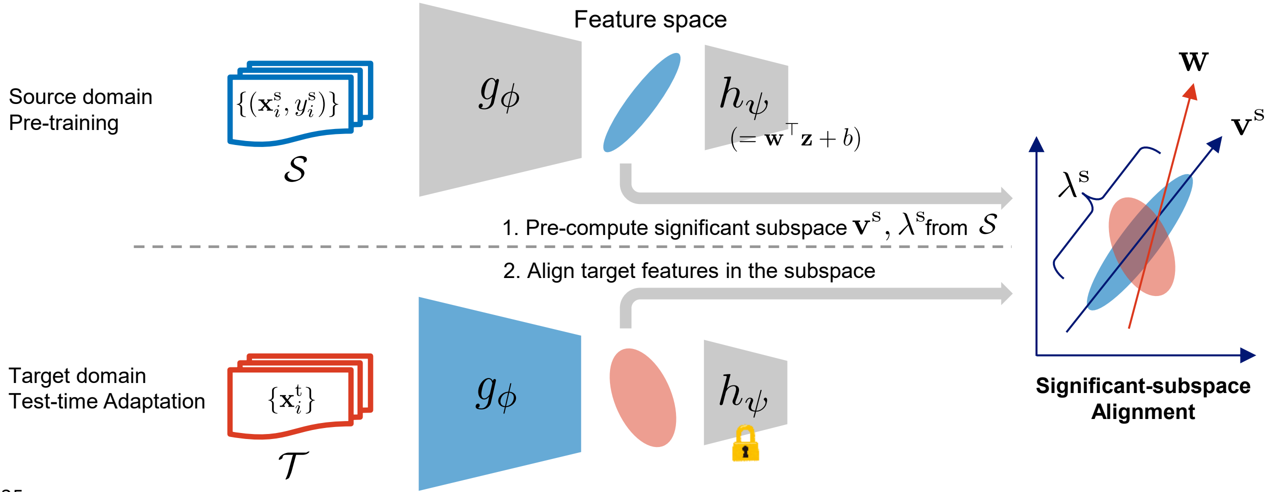
- Features are less diverse in regression than classification [1]
- <u>**Our finding**: Features are distributed only in a small subspace in regression models</u>
  → Most feature dimensions do not affect the output

Dataset	#Subspace dims.
SVHN	14
UTKFace	76
Biwi Kinect (mean)	34.5
California Housing (100 dims.)	40

[1] Zhang et al., Improving Deep Regression with Ordinal Entropy, ICLR 2023.

Proposed Method

- **Approach**
  - **Subspace detection**: Detects the representative source feature subspace using PCA
  - **Dimension weighting**: Weights the subspace dimensions based on the significant to the output



Experiment

- Compared TTA performance on regression tasks
  - SVHN-MNIST: Directly predicts a scalar value of the label
    - Source: SVHN, Target: MNIST
  - California Housing: Predicts housing prices (tabular data)
    - Source: non-coastal area, Target: coastal area
  - UTKFace: Predicts the age of the person in a face image
    - Source: clean image, Target: corrupted image

SVHN-MNIST

UTKFace

Result

NTT

- Our method consistently had higher performance
  - Baseline methods sometimes had lower performance than Source (no adaptation)

SVHN-MNIST

Method	$R^2 (\uparrow)$	RMSE ($\downarrow$)	MAE ($\downarrow$)
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	$0.874_{\pm 0.00}$	$1.028_{\pm 0.00}$	$0.575_{\pm 0.00}$

California Housing

Method	$R^2 (\uparrow)$	RMSE ($\downarrow$)	MAE ($\downarrow$)
Source	0.605	0.684	0.516
BN-adapt	$0.318_{\pm 0.00}$	$0.899_{\pm 0.00}$	$0.699_{\pm 0.00}$
Prototype	$-0.726_{\pm 0.01}$	$1.431_{\pm 0.00}$	$1.196_{\pm 0.00}$
FR	$0.510_{\pm 0.01}$	$0.762_{\pm 0.01}$	$0.534_{\pm 0.01}$
RSD	-	-	-
SSA (ours)	$\mathbf{0.639}_{\pm 0.00}$	$\mathbf{0.655}_{\pm 0.00}$	$\mathbf{0.469}_{\pm 0.00}$
Oracle	$0.729_{\pm 0.00}$	$0.567_{\pm 0.00}$	$0.404_{\pm 0.00}$

Result

NTT

- Our method consistently had higher performance
  - Baseline methods sometimes had lower performance than Source (no adaptation)

UTKFace (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

Thank you for watching!

Paper (OpenReview)

GitHub