Covariate-Shift Robust and Feature-wise Adaptive Transfer Learning for High-Dimensional Regression

PennState

Zelin He, Ying Sun, Jingyuan Liu, Runze Li

Introduction

In **transfer learning**, we observe

- A **Target sample:** $(\boldsymbol{X}^{(0)}, \boldsymbol{y}^{(0)}) \sim P^{(0)}(\boldsymbol{x}, y) = P^{(0)}(y|\boldsymbol{x})P^{(0)}(\boldsymbol{x})$.
- Multiple **Source samples:** for $k = 1, \ldots, K$, $(\boldsymbol{X}^{(k)}, \boldsymbol{y}^{(k)}) \sim P^{(k)}(\boldsymbol{x}, y) = P^{(k)}(y|\boldsymbol{x})P^{(k)}(\boldsymbol{x})$.

Our Goal is to learn the target model $P^{(0)}(y|\boldsymbol{x})$, by incorporating source information.

Challenge 1: covariate shift $P^{(k)}(\boldsymbol{x}) \neq P^{(0)}(\boldsymbol{x})$.

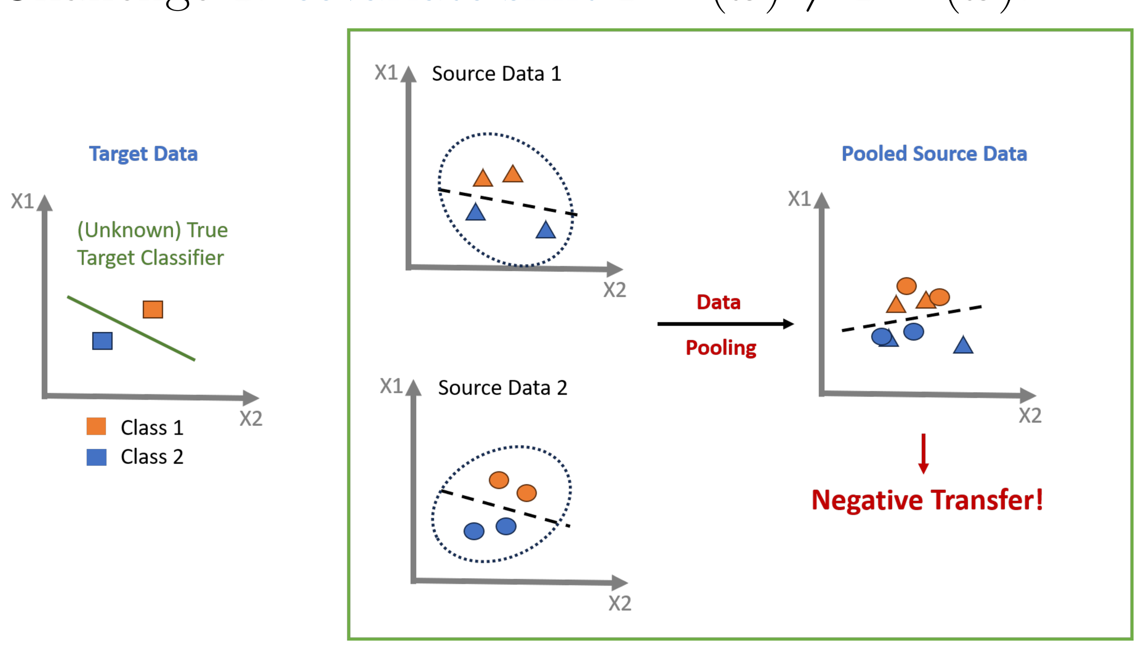


Figure 1: How failure to manage covariate shifts across sources can result in negative transfer.

⇒**Our first question:** *How to develop a computationally efficient method that handles model shift, while being robust to covariate shift?*

Challenge 2: model shift $P^{(k)}(\boldsymbol{x}, y) \neq P^{(0)}(\boldsymbol{x}, y)$.

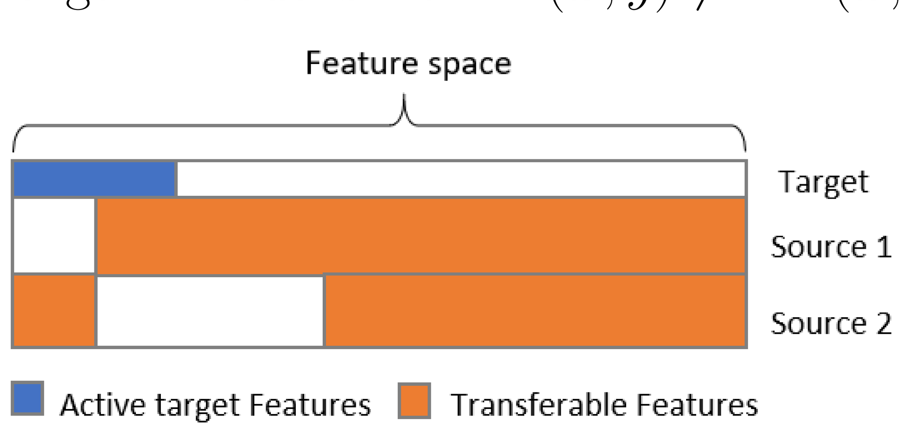


Figure 2: Illustration of feature-wise model shift patterns

⇒**Our second question:** *How to adapt to the high-dimensional feature-wise model shift from each source during knowledge transfer?*

Problem Setting

High-dimensional Linear Regression:
Sample-level target model (with sample size n_T):

$$\boldsymbol{y}^{(0)} = \boldsymbol{X}^{(0)}\boldsymbol{\beta}^{(0)} + \boldsymbol{\epsilon}^{(0)},$$

Sample-level source model (with sample size n_S):

$$\boldsymbol{y}^{(k)} = \boldsymbol{X}^{(k)}(\boldsymbol{\beta}^{(0)} + \boldsymbol{\delta}^{(k)}) + \boldsymbol{\epsilon}^{(k)}$$

- $E(\boldsymbol{\epsilon}^{(k)}) = \boldsymbol{0}$, $\mathrm{Cov}(\boldsymbol{\epsilon}^{(k)}) = \sigma^2\boldsymbol{I}$, $\boldsymbol{\epsilon}^{(k)} \perp\!\!\!\perp \boldsymbol{X}^{(k)}$
- $\boldsymbol{\beta}^{(0)} \in \mathbb{R}^p$ is high-dimensional yet sparse.
- **Covariate shift**: $\mathrm{Cov}(\boldsymbol{X}_i^{(k)}) = \boldsymbol{\Sigma}^{(k)}$ varies.
- **Model shift**: $\boldsymbol{\delta}^{(k)} \in \mathbb{R}^p$ varies across $k \in [K]$.

Key: Fused-Regularizer

We achieve transfer learning by solving

$$\underset{\boldsymbol{\beta},\boldsymbol{\delta}}{\operatorname{argmin}}\{(2N)^{-1}\sum_{k=0}^{K}\|\boldsymbol{y}^{(k)} - \boldsymbol{X}^{(k)}(\boldsymbol{\beta}^{(0)} + \boldsymbol{\delta}^{(k)})\|_2^2 + \underbrace{\lambda_0\sum_{j=1}^{p}\hat{w}_{0j}|\boldsymbol{\beta}_j^{(0)}|}_{\text{Sparsify}} + \underbrace{\lambda_1\sum_{k=1}^{K}\sum_{j=1}^{p}\hat{w}_{kj}|\boldsymbol{\delta}_j^{(k)}|}_{\text{Transfer}}\}, \quad (1)$$

- The first term measures the average fitness.
- The fused-regularizer achives sparsity of $\boldsymbol{\beta}^{(0)}$ and shrinking the contrast $\boldsymbol{\delta}^{(k)}$ for transfer.
- The weight adjusts the info transfer from $\boldsymbol{\delta}_j^{(k)}$.

Why it is covariate-shift robust? It adjusts for the kth source's shift, $\boldsymbol{\delta}^{(k)}$, by separately estimating it using the source-specific sample $(\boldsymbol{X}^{(k)}, \boldsymbol{y}^{(k)})$.
Why it is feature-wise adaptive? It adjusts weights, w_{kj}, applied to each $\boldsymbol{\delta}_j^{(k)}$:

- apply stronger penalties to transferable features with negligible $\boldsymbol{\delta}_j^{(k)}$;
  → shrink $\boldsymbol{\delta}_j^{(k)}$ to 0, i.e. pool $\boldsymbol{\beta}_j^{(k)}$ and $\boldsymbol{\beta}_j^{(0)}$, if the j-th feature from the k-th source is transferable.
- prevents excessive penalties to non-transferable features with large $\boldsymbol{\delta}_j^{(k)}$.
  → prevent introducing bias from model shifts.

Theory: Robustness

Consider the parameter space

$$\Theta(s,h) = \{\boldsymbol{\beta}^{(0)}, \boldsymbol{\delta} : \|\boldsymbol{\beta}^{(0)}\|_0 \le s, \|\boldsymbol{\delta}^{(k)}\|_1 \le h_k\}.$$

We first propose an unweighted two-step method with the fused-regularizer, named TransFusion, which under mild conditions, w.h.p. yields

$$\|\hat{\boldsymbol{\beta}}_{\mathrm{TF}}^{(0)} - \boldsymbol{\beta}^{(0)}\|_2^2 \lesssim \underbrace{\frac{s\log p}{Kn_S + n_T}}_{\text{Estimate } \boldsymbol{\beta}^{(0)}} + \underbrace{\bar{h}\sqrt{\frac{\log p}{n_T}} \wedge \bar{h}^2}_{\text{Correct } \boldsymbol{\delta}^{(k)}\text{s}}.$$

Baseline: TransLasso, which adopts a "pooling pertraining + debiasing" strategy, yields

$$\|\hat{\boldsymbol{\beta}}_{\mathrm{Baseline}}^{(0)} - \boldsymbol{\beta}^{(0)}\|_2^2 \lesssim \frac{s\log p}{Kn_S + n_T} + C_{\Sigma}\bar{h}\sqrt{\frac{\log p}{n_T}} \wedge \bar{h}^2,$$

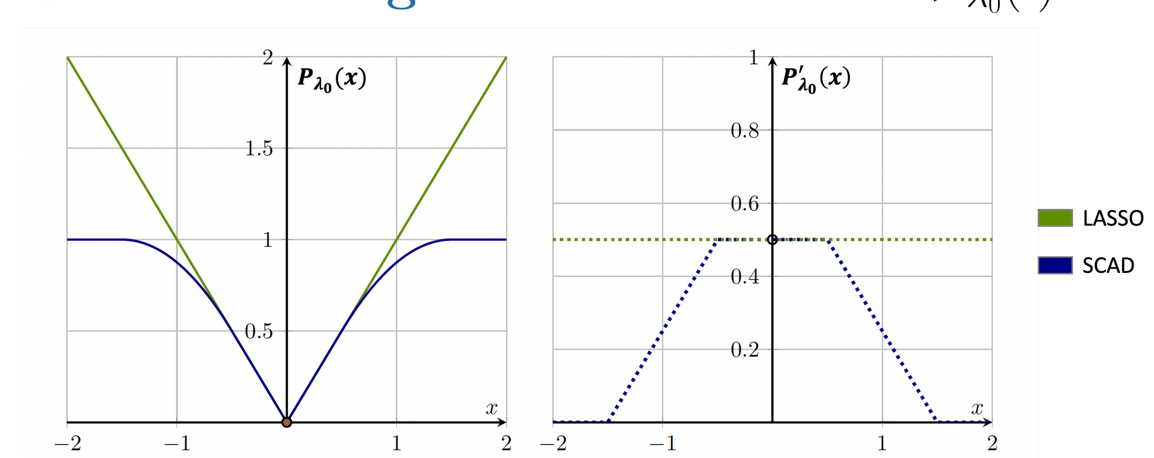
where C_{Σ} measures the covariate-shift strength:

$$C_{\Sigma} := 1 + \max_{j\le p}\max_{k}\left\| e_j^{\top}(\boldsymbol{\Sigma}^{(k)} - \boldsymbol{\Sigma}^{(0)})\left(\sum_{1\le k\le K}\frac{1}{K}\boldsymbol{\Sigma}^{(k)}\right)^{-1}\right\|_1,$$

and can diverge in the order of $O(\sqrt{p})$!

Theory: Adaptation

Choice of weight: **folded-concave $\mathcal{P}_{\lambda_0}(\cdot)$.**



Borrowing the idea of local linear approximation, take $\hat{w}_{0j} \propto \mathcal{P}'_{\lambda_0}(\hat{\boldsymbol{\beta}}_{\mathrm{init},j}^{(0)})$ and $\hat{w}_{kj} \propto \mathcal{P}'_{\lambda_0}(\hat{\boldsymbol{\delta}}_{\mathrm{init},j}^{(k)})$, where $\hat{\boldsymbol{\beta}}_{\mathrm{init},j}^{(0)}$ and $\hat{\boldsymbol{\delta}}_{\mathrm{init},j}^{(k)}$ are initial estimators of $\boldsymbol{\beta}_j^{(0)}$ and $\boldsymbol{\delta}_j$.

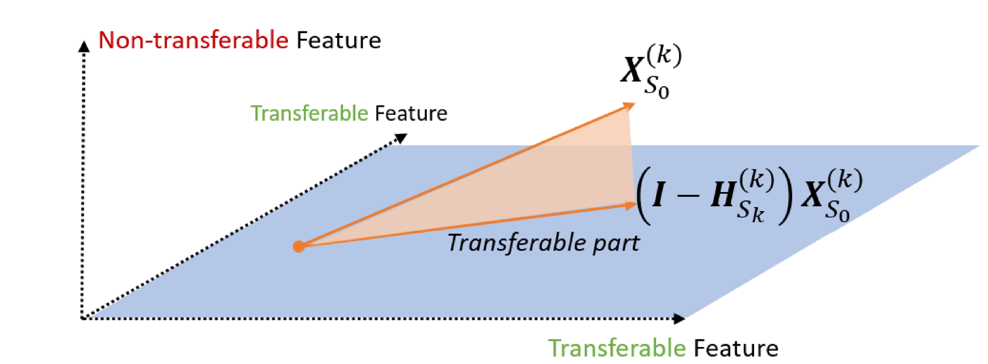
1. Define **sparsity structure:**
   - Active target feature set: $S_0 = \{j : \boldsymbol{\beta}_j^{(0)} \neq 0\}$,
   - Inactive target feature set: $S_0 = \{j : \boldsymbol{\beta}_j^{(0)} = 0\}$;
2. Define **transferability structure:**
   - Non-transferable set: $S_k = \{j : \boldsymbol{\delta}_j^{(k)} \neq 0\}$, $k = 1, \ldots, K$,
   - Transferable set: $S_k^c = \{j : \boldsymbol{\delta}_j^{(k)} = 0\}$, $k = 1, \ldots, K$.

Theory: Adaptation (Cont'd)

Under mild conditions, if the transferable structure is detectable, solving (1) yields the oracle solution

$$\hat{\boldsymbol{\beta}}_{\mathrm{ora},S_0}^{(0)} = [\tilde{\boldsymbol{X}}_{S_0}^{\top}\tilde{\boldsymbol{X}}_{S_0}]^{-1}\tilde{\boldsymbol{X}}_{S_0}^{\top}\boldsymbol{y} \quad \text{and} \quad \hat{\boldsymbol{\beta}}_{\mathrm{ora},S_0^c}^{(0)} = \boldsymbol{0}.$$

- $\tilde{\boldsymbol{X}}_{S_0} = ((\boldsymbol{X}_{S_0}^{(0)})^{\top}, (\tilde{\boldsymbol{X}}_{S_0}^{(1)})^{\top}, \ldots, (\tilde{\boldsymbol{X}}_{S_0}^{(K)})^{\top})^{\top}$.
- $\tilde{\boldsymbol{X}}_{S_0}^{(k)} = (\boldsymbol{I} - \mathbf{H}_{S_k}^{(k)})\boldsymbol{X}_{S_0}^{(k)}$: the projection of the active target feature onto the null space of the non-transferable feature in the k-th source.



Real-world Evidence

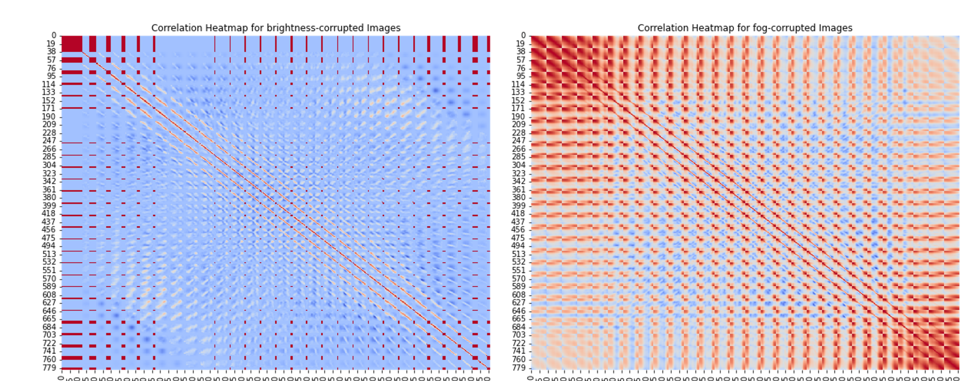


Figure 3: Covariate shifts in C-MNIST dataset: images with different contamination demonstrate distinct pixel correlations.

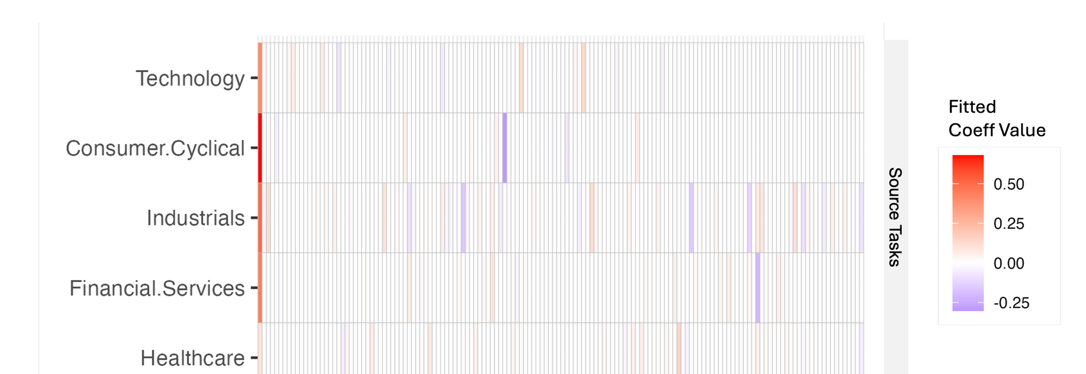


Figure 4: Feature-wise model shifts in financial data: stocks across sectors differ in key accounting metric features.

Our method demonstrates favorable performance over other approaches in both datasets.