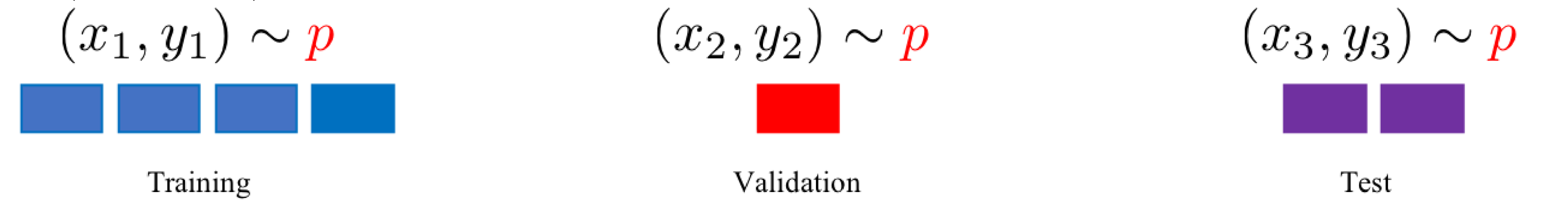
Towards Accurate Model Selection in Deep Unsupervised Domain Adaptation

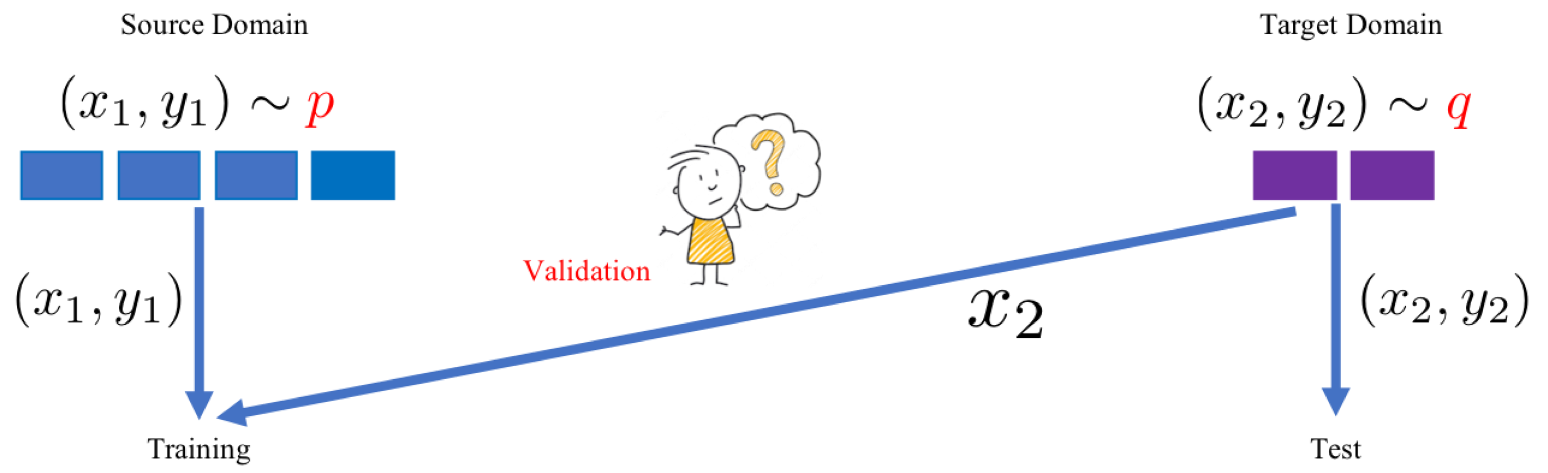
Kaichao You, Ximei Wang, Mingsheng Long, Michael I. Jordan

Validation in UDA: the problem

- Supervised Learning
  - train/validation/test data come from the same distribution



- Unsupervised Domain Adaptation (UDA)
  - train/test data come from different distributions
  - test data is unlabeled until the test phase, so target labels are not available for validation



- Status quo of model selection in UDA
  - **Source Risk**: a highly biased estimator of the underlying target risk in UDA
  - **Target Risk**: requires target labels that contradicts with the assumption of UDA
  - **IWCV** unstable because of the unbounded variance
  - **TrCV**: requires target labels that contradicts with the assumption of UDA

Method	Working Assumptions		Technical Advantages	
	covariate shift	w/o target labels	unbiased	controlled variance
Source Risk	✗	✓	✗	✗
Target Risk	✓	✗	✓	✓
IWCV	✓	✓	✓	✗
TrCV	✓	✗	✓	✗
DEV (Proposed)	✓	✓	✓	✓

IWCV: the previous solution

- Covariate Shift Assumption $p(y|\mathbf{x}) = q(y|\mathbf{x})$
- Model Selection: estimate Target Risk $\mathcal{R}(g) = \mathbb{E}_{\mathbf{x}\sim q}\ell(g(\mathbf{x}), y)$
- Importance Weighted Cross Validation

$$\begin{aligned}\mathbb{E}_{\mathbf{x}\sim p} w(\mathbf{x})\ell(g(\mathbf{x}), y) &= \mathbb{E}_{\mathbf{x}\sim p}\frac{q(\mathbf{x})}{p(\mathbf{x})}\ell(g(\mathbf{x}), y)\\ &= \int_p \frac{q(\mathbf{x})}{p(\mathbf{x})}\ell(g(\mathbf{x}), y)p(\mathbf{x})\mathrm{d}\mathbf{x}\\ &= \int_q \ell(g(\mathbf{x}), y)q(\mathbf{x})\mathrm{d}\mathbf{x}\\ &= \mathbb{E}_{\mathbf{x}\sim q}\ell(g(\mathbf{x}), y)\\ &= \mathcal{R}(g),\end{aligned}$$

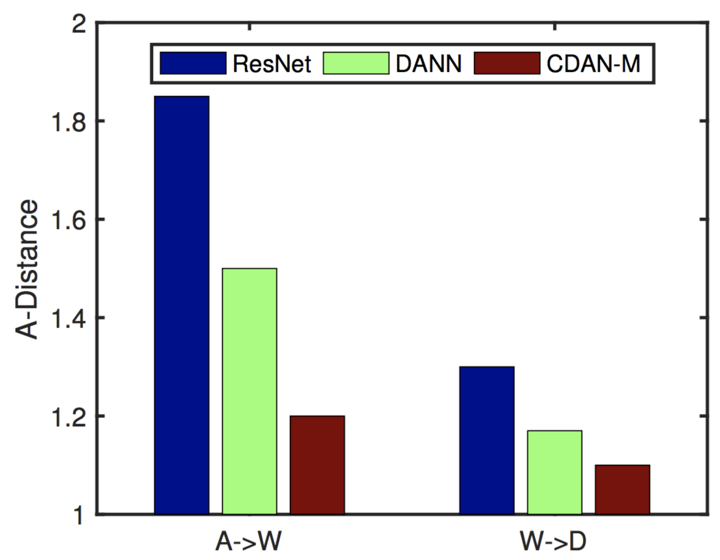
- Problems in IWCV:
  - Unbiased but the variance is unbounded
    $\mathrm{Var}_{\mathbf{x}\sim p}[\ell_w] \le d_{\alpha+1}(q\|p)\,\mathcal{R}(g)^{1-\frac{1}{\alpha}} - \mathcal{R}(g)^2.$
    $d_\alpha(p\|q) = 2^{D_\alpha(p\|q)} = \left[\sum_x \frac{p^\alpha(x)}{q^{\alpha-1}(x)}\right]^{\frac{1}{\alpha-1}}$ (Rényi Divergence)
  - Density ratio is not readily accessible
    Fitting a gaussian distribution as in the original paper is not reasonable.

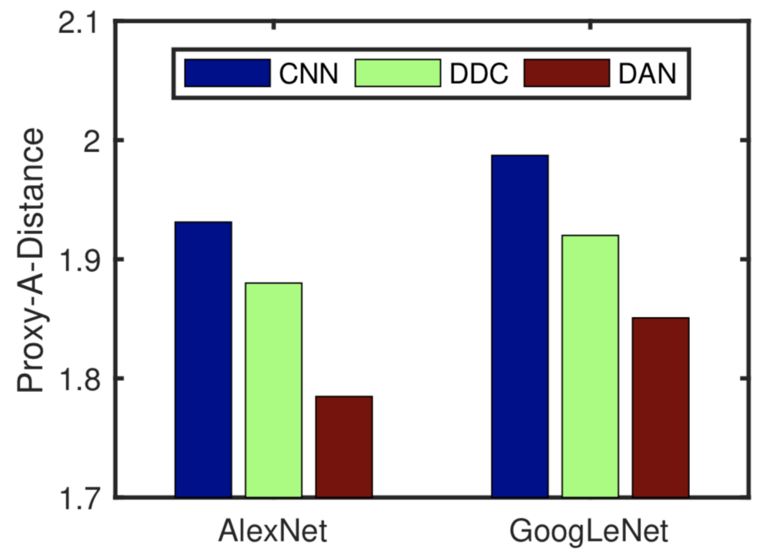
Insights

- Domain adaptation reduces distribution discrepancy, thus lowering the variance upper-bound
- Use a control variate to explicitly reduce the variance
- Density ratio can be estimated discriminatively

Embed Adapted Features into Model Selection

- Recent feature adaptation methods reduce distribution discrepancy
  - $d_{\alpha+1}(q_f\|p_f) \le d_{\alpha+1}(q\|p)$





Control Variate

- $\mathbb{E}[z] = \zeta, \mathbb{E}[t] = \tau$
- $z^\star = z + \eta(t - \tau).$
- $\mathbb{E}[z^\star] = \mathbb{E}[z] + \eta\mathbb{E}[t - \tau] = \zeta + \eta(\mathbb{E}[t] - \mathbb{E}[\tau]) = \zeta.$
- $\mathrm{Var}[z^\star] = \mathrm{Var}[z + \eta(t - \tau)] = \eta^2\mathrm{Var}[t] + 2\eta\mathrm{Cov}(z, t) + \mathrm{Var}[z]$
- $\min \mathrm{Var}[z^\star] = (1 - \rho_{z,t}^2)\mathrm{Var}[z]$, when $\hat{\eta} = -\frac{\mathrm{Cov}(z,t)}{\mathrm{Var}[t]}$
- w_f can be used as a control variate
  $\mathbb{E}_{\mathbf{x}\sim p_f} w_f(\mathbf{x}) = \mathbb{E}_{\mathbf{x}\sim p_f}\frac{q_f(\mathbf{x})}{p_f(\mathbf{x})} = \int \frac{q_f(\mathbf{x})}{p_f(\mathbf{x})}p_f(\mathbf{x})\mathrm{d}\mathbf{x} = 1.$

Discriminative Density Ratio Estimation

- Can be estimated by a discriminative model to distinguish source examples from target examples

$$\begin{aligned}w_f(\mathbf{x}) = \frac{q_f(\mathbf{x})}{p_f(\mathbf{x})} &= \frac{J_f(\mathbf{x}|d=0)}{J_f(\mathbf{x}|d=1)} = \frac{J_f(d=1)\,J_f(\mathbf{x})J_f(d=0|\mathbf{x})}{J_f(d=0)\,J_f(\mathbf{x})J_f(d=1|\mathbf{x})}\\ &= \frac{J_f(d=1)\,J_f(d=0|\mathbf{x})}{J_f(d=0)\,J_f(d=1|\mathbf{x})} = \frac{n_s J_f(d=0|\mathbf{x})}{n_t J_f(d=1|\mathbf{x})},\end{aligned}$$

Algorithm in Detail

Algorithm 2 Deep Embedded Validation (DEV)

Input: A set of candidate models $G_m = \{g_i(\mathbf{x})\}_{i=1}^m$
Output: The best model $(G_m)_{\hat{i}}$

Get DEV Risks of all models $\mathcal{R} = \{\mathrm{GetRisk}(g_i)\}_{i=1}^m$
Rank the best model $\hat{i} = \arg\min_{1\le i\le m}\mathcal{R}_i$

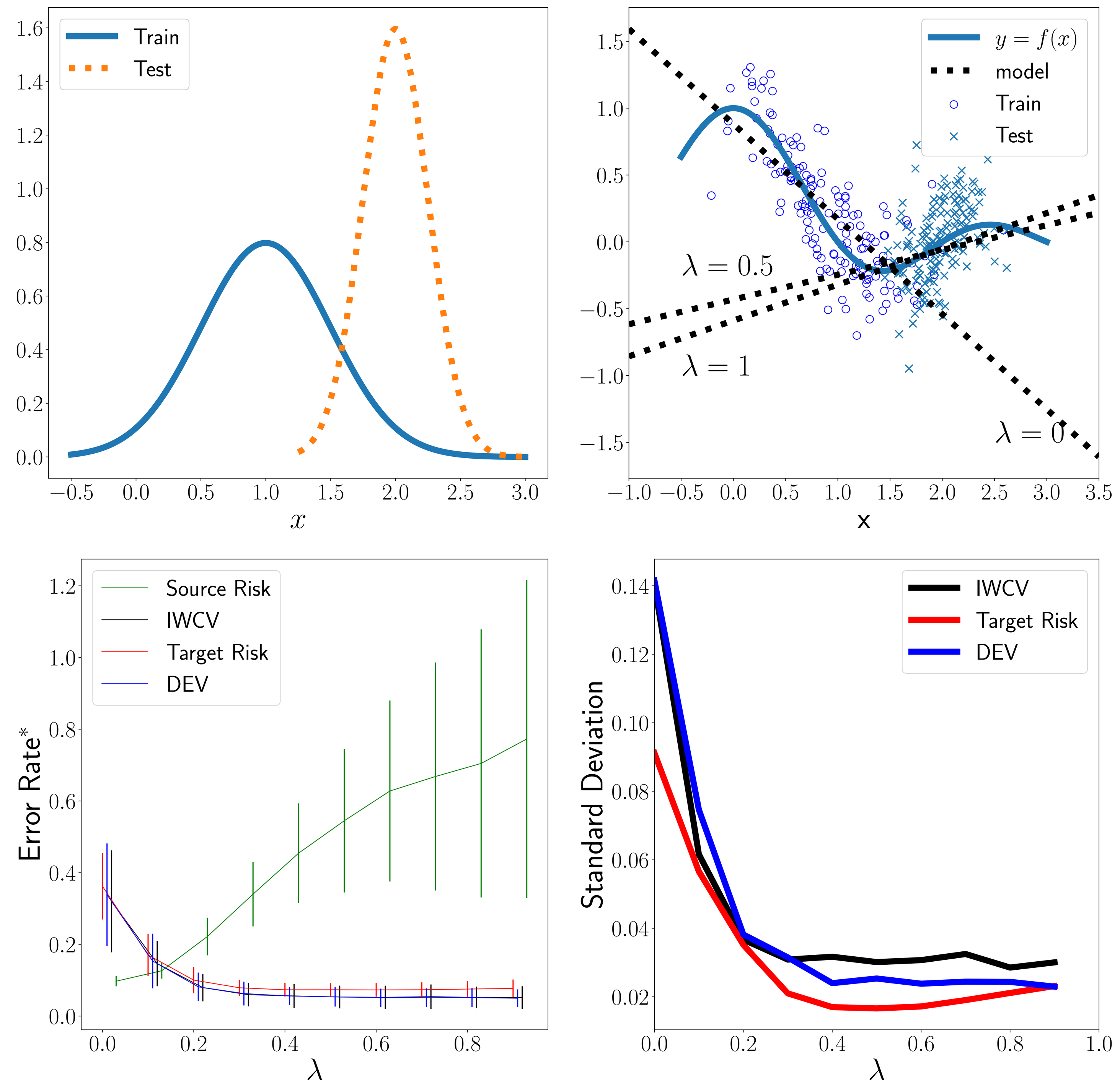
Algorithm 1 GetRisk

Input: Candidate model $g(\mathbf{x}) = \mathcal{T}(F(\mathbf{x}))$
Training set $\mathcal{D}_{\mathrm{tr}} = \{(\mathbf{x}_i^{\mathrm{tr}}, y_i^{\mathrm{tr}})\}_{i=1}^{n_{\mathrm{tr}}}$
Validation set $\mathcal{D}_{\mathrm{v}} = \{(\mathbf{x}_i^{\mathrm{v}}, y_i^{\mathrm{v}})\}_{i=1}^{n_{\mathrm{v}}}$
Test set $\mathcal{D}_{\mathrm{ts}} = \{(\mathbf{x}_i^{\mathrm{ts}})\}_{i=1}^{n_{\mathrm{ts}}}$
$\mathcal{D}_s$ is partitioned into $\mathcal{D}_{\mathrm{tr}}$ and $\mathcal{D}_{\mathrm{v}}$
Output: DEV Risk $\mathcal{R}_{\mathrm{DEV}}(g)$ of model g

Compute features and predictions using model g:
$\mathcal{F}_{\mathrm{tr}} = \{\boldsymbol{f}_i^{\mathrm{tr}}\}_{i=1}^{n_{\mathrm{tr}}}, \mathcal{F}_{\mathrm{ts}} = \{\boldsymbol{f}_i^{\mathrm{ts}}\}_{i=1}^{n_{\mathrm{ts}}}$
$\mathcal{F}_{\mathrm{v}} = \{\boldsymbol{f}_i^{\mathrm{v}}\}_{i=1}^{n_{\mathrm{v}}}, \mathcal{Y}_{\mathrm{v}} = \{\hat{y}_i^{\mathrm{v}}\}_{i=1}^{n_{\mathrm{v}}}$
Train a two-layer logistic regression model M to classify $\mathcal{F}_{\mathrm{tr}}$ and $\mathcal{F}_{\mathrm{ts}}$ (label $\mathcal{F}_{\mathrm{tr}}$ as 1 and $\mathcal{F}_{\mathrm{ts}}$ as 0)
Compute $w_{\boldsymbol{f}}(\mathbf{x}_i^{\mathrm{v}}) = \frac{n_{\mathrm{tr}}}{n_{\mathrm{ts}}}\frac{1-M(\boldsymbol{f}_i^{\mathrm{v}})}{M(\boldsymbol{f}_i^{\mathrm{v}})}$, $W = \{w_{\boldsymbol{f}}(\mathbf{x}_i^{\mathrm{v}})\}_{i=1}^{n_{\mathrm{v}}}$
Compute weighted loss $L = \{w_{\boldsymbol{f}}(\mathbf{x}_i^{\mathrm{v}})\ell(\hat{y}_i^{\mathrm{v}}, y_i^{\mathrm{v}})\}_{i=1}^{n_{\mathrm{v}}}$
Estimate coefficient $\eta = -\frac{\widehat{\mathrm{Cov}}(L,W)}{\widehat{\mathrm{Var}}[W]}$
Compute DEV Risk:
$\mathcal{R}_{\mathrm{DEV}}(g) = \mathrm{mean}(L) + \eta\,\mathrm{mean}(W) - \eta$

Experimental Results

- Experiments on a toy problem under covariate shift



- Experiments on real-world problems
  - Various datasets: VisDA/Office/Digits
  - Various models: CDAN, MCD, GTA
  - Deep Embedded Validation is empirically validated

School of Software - Tsinghua University - China National Engineering Lab for Big Data Software University of California, Berkeley Mail: youkaichao@gmail.com