Transfer Neural Trees for Heterogeneous Domain Adaptation

Wei–Yu Chen[1,2], Tzu–Ming Hsu[2], Yao–Hung Tsai[3], Yu–Chiang Frank Wang[2], Ming–Syan Chen[1]
[1]Graduate Institute of Electrical Engineering, National Taiwan University, Taipei, Taiwan
[2]Research Center for Information Technology Innovation, Academia Sinica, Taipei, Taiwan
[3]Department of Machine Learning, Carnegie Mellon University, Pittsburgh, USA

Introduction

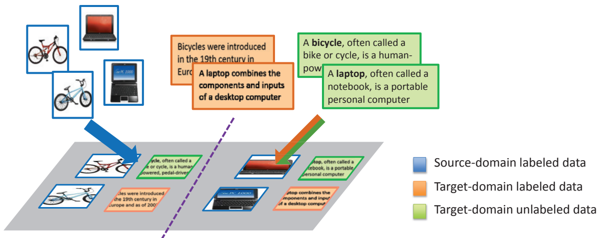
- **Domain adaptation**:
  Address the same learning task across different domains
- **Heterogeneous domain adaptation**:
  Source and target-domain data are described by distinct types of features.

Related Works

- Map cross-domain data onto a common subspace for classification
- ✓ Jointly learning of mapping & classification functions from cross-domain labeled data [1]
- ✓ Project cross-domain data for suppressing domain differences [2]
- ✓ A pair of DNNs with shared parameters for matching cross-domain data. [3]

Highlights

- Semi-supervised learning of feature mapping and classification functions from cross-domain labeled & unlabeled data
- No cross-domain data pairs are required.
- A shared prediction layer for joint adaptation & classification

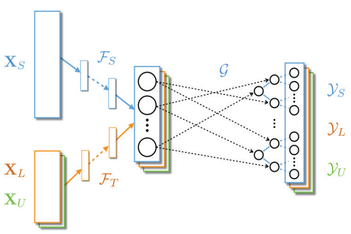


Proposed Method

Transfer Neural Trees (TNT)

- Source-domain mapping $\mathcal{F}_S$
- Target-domain mapping $\mathcal{F}_T$
- Prediction layer $\mathcal{G}$

$D_S = [X_S, \mathcal{Y}_S]$: Labeled source-domain data
$D_L = [X_L, \mathcal{Y}_L]$: Labeled target-domain data
$D_U = [X_U, \mathcal{Y}_U]$: Unlabeled target-domain data



Learning $\mathcal{F}_S$ and $\mathcal{G}$:

✓ Minimize prediction loss L_p of source-domain data X_S

$$\min_{\mathcal{F}_S,\mathcal{G}} \sum_{(x,y)\in D_S} L_p(\Theta,\boldsymbol{\pi};\mathbf{x},y) \text{ where } L_p(\Theta,\boldsymbol{\pi};\mathbf{x},y) = \sum_{T\in F} -\log\big(P_T[y|\mathbf{x},\Theta,\boldsymbol{\pi}]\big)$$

Learning $\mathcal{F}_T$ with fixed $\mathcal{G}$ (semi-supervised learning):

✓ Minimize prediction loss L_p of target-domain labeled data X_L
✓ Minimize embedding loss L_e of target-domain labeled & unlabeled data

$$\min_{\mathcal{F}_T} \sum_{(x,y)\in D_L} L_p(\Theta,\boldsymbol{\pi};\mathbf{x},y) + \lambda \sum_{\tilde{y}} \sum_{\mathbf{x}\in\{X_L,X_U\}} L_e(\Theta,\boldsymbol{\pi};\mathbf{x},\tilde{y})$$

$$\text{where } L_e(\Theta,\boldsymbol{\pi};\mathbf{x},\tilde{y}) = \sum_{\mathbf{x}\in\{X_L,X_U\},T\in F} -P_T[\tilde{y}|\mathbf{x},\Theta,\boldsymbol{\pi}]\,\frac{P_F[\tilde{y}|\mathbf{x},\Theta,\boldsymbol{\pi}]}{P_F[\tilde{y}|\Theta,\boldsymbol{\pi}]},$$

$$P_F[\tilde{y}|\Theta,\boldsymbol{\pi}] = \frac{1}{n_l+n_u}\sum_{\mathbf{x}\in\{X_L,X_U\}} -P_T[\tilde{y}|\mathbf{x},\Theta,\boldsymbol{\pi}]$$

• Increase prediction consistency → preserve structural consistency btw X_L & X_U

Prediction layer $\mathcal{G}$

✓ **Deep Neural Decision Forest (dNDF)** [4]: back-propagatable random forest

Probability of reaching leaf node l:

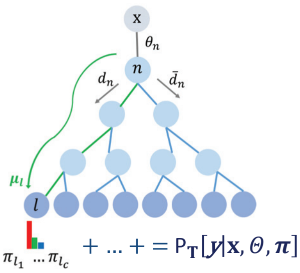
$$\mu_l(\mathbf{x}|\Theta) = \prod_{n\in N} d_n(\mathbf{x};\Theta)^{1_{l\swarrow n}}\bar{d}_n(\mathbf{x};\Theta)^{1_{l\searrow n}}$$

Class-label distribution at leaf node:

$$\pi_{l_y}{}^{(t+1)} = \frac{1}{Z_l^{(t)}}\Big(\sum_{(\mathbf{x},y')\in D_S} \frac{1_{y=y'}\pi_{l_y}{}^{(t)}\mu_l(\mathbf{x}|\Theta)}{P_T[y|\mathbf{x},\Theta,\boldsymbol{\pi}^{(t)}]}\Big)$$

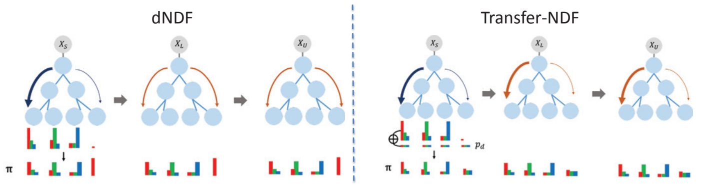
Overall prediction:

$$P_T[y|\mathbf{x},\Theta,\boldsymbol{\pi}] = \sum_{l\in L}\pi_{l_y}\mu_l(\mathbf{x}|\Theta)$$



✓ **Transfer Neural Decision Forest (Transfer-NDF)**:

dNDF with *stochastic pruning*: $\pi_{l_y}{}^{(t+1)} = \frac{1}{Z_l^{(t)}}\Big(p_d + \sum_{(\mathbf{x},y')\in D_S} \frac{1_{y=y'}\pi_{l_y}{}^{(t)}\mu_l(\mathbf{x}|\Theta)}{P_T[y|\mathbf{x},\Theta,\boldsymbol{\pi}^{(t)}]}\Big)$



• Automatically prune the leaf nodes with insufficient adaptation abilities

Experiments

Datasets

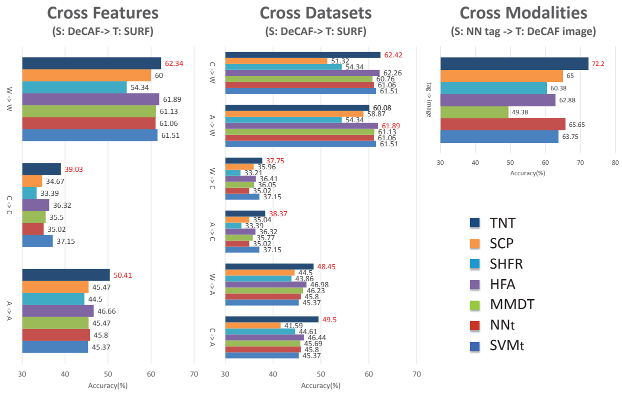
✓ **Object recognition (10 classes) :**
– Amazon: 959 DeCAF/SURF features
– Webcam: 296 DeCAF/SURF features
– Caltech: 1124 DeCAF/SURF features

✓ **Text-to-image recognition (8 classes) :**
– NUS-WIDE tag data: 800 NN features
– ImageNet image: 800 DeCAF features

Settings

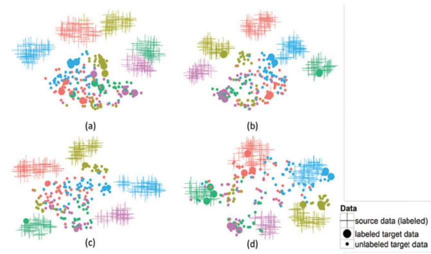
– Source domain: all data in dataset as X_S
– Target domain: 3 per class as X_L, the rest as X_U

Evaluation



Visualization

✓ Different $\mathcal{G}$ in TNT



S -> T	(a) NN	(b) NDF	(c) Transfer-NDF	(d) Transfer-NDF + embedding loss
C-> C	32.31	33.94	34.62	39.03

Conclusions

- TNT for semi-supervised & cross-domain deep learning
- Transfer-NDF with stochastic pruning for HDA
- Embedding loss in TNT for preserving prediction & structural consistency
- Promising results on cross-feature, domain, and modality classification tasks

Reference

[1] Hoffman et al. "Efficient learning of domain-invariant image representations," ICLR, 2013.
[2] Wang et al. "Heterogeneous domain adaptation using manifold alignment," IJCAI, 2011.
[3] Shu et al. Weakly-shared deep transfer networks for heterogeneous-domain knowledge propagation," ACMMM, 2015.
[4] Kontschieder et al. "Deep Neural Decision Forests," ICCV. 2015.