ParNeC 模式识别与神经计算研究组
PAttern Recognition and NEural Computing

A Brief Introduction of Model Reuse

报告人：唐英鹏

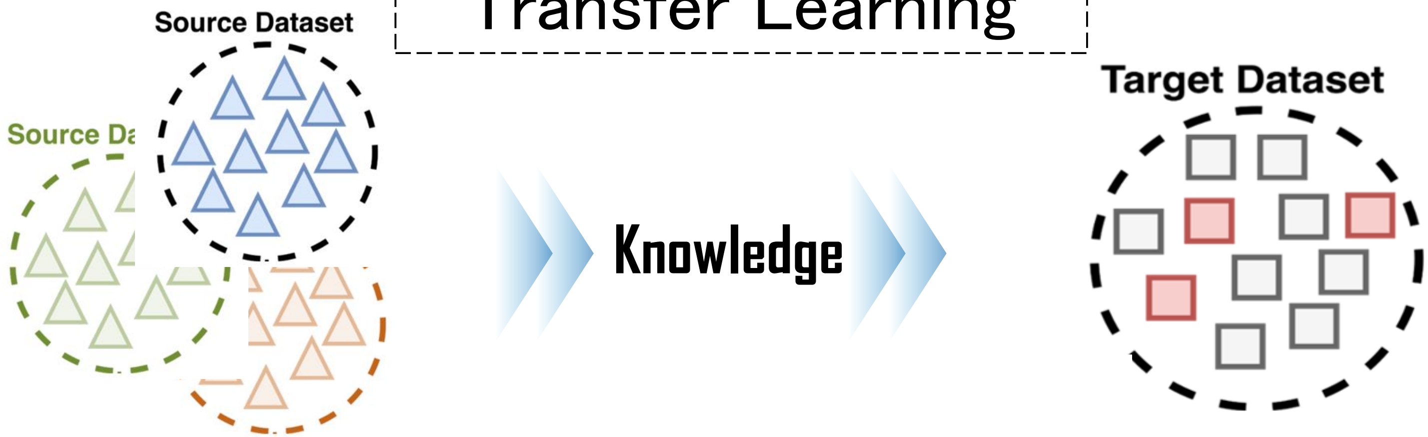
2021-4-14

Contents

模式识别与神经计算研究组
PAttern Recognition and NEural Computing

Background

Transfer Learning



Model Reuse

- Model reuse (also called learning from auxiliary classifiers, hypothesis transfer learning) aim at reusing pre-trained models to help related learning tasks.

Existing methods

ParNeC 模式识别与神经计算研究组 PAttern Recognition and NEural Computing

- Updating the pre-trained model on the current task, like fine-tuning neural networks.

- Training a new model with the help of source models, like biased regularization. Or select source models properly for prediction.

How to **learn the target model?**

- Biased regularization (popular!)
- Select source models for prediction (Arxiv' 20)

How to **rate the source models?**

1. According to the performance on the labeled target data (TPAMI' 14, ML' 20)
2. Require additional information (e.g., distribution gap) (Arxiv' 20)
3. Exploit the unlabeled data (e.g., the semi-supervised metrics) (submit to ICML)

Contents

Fast Rates by Transferring from Auxiliary Hypotheses

Ilja Kuzborskij[1] · Francesco Orabona[2]

1 Idiap Research Institute, Centre du Parc, Rue Marconi 19, 1920 Martigny, Switzerland

2 Department of Computer Science, Stony Brook University, Stony Brook, NY 11794, USA

ML’17

- Biased Regularized Least Squares

$$\hat{\mathbf{w}} = \underset{\mathbf{w}\in\mathcal{H}}{\operatorname{argmin}} \left\{ \frac{1}{m} \sum_{i=1}^{m} (\langle \mathbf{w}, \mathbf{x}_i \rangle - y_i)^2 + \lambda \left\| \mathbf{w} - \mathbf{W}^{\mathrm{src}} \boldsymbol{\beta} \right\|_2^2 \right\}.$$

where

training set $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^{m}$

target hypothesis $h(\mathbf{x}) = \langle \hat{\mathbf{w}}, \mathbf{x} \rangle$

source hypotheses $\{\mathbf{w}_i^{\mathrm{src}}\}_{i=1}^{n} \subset \mathcal{H}$

$\boldsymbol{\beta} \in \mathbb{R}^n$ and $\lambda \in \mathbb{R}_+$

Introduce $\mathbf{w}'$, such that $\mathbf{w}' = \mathbf{w} - \mathbf{W}^{\mathrm{src}} \boldsymbol{\beta}$. Then we have that problem (3) is equivalent to

Target hypothesis

$$\min_{\mathbf{w}\in\mathcal{H}} \left\{ \frac{1}{m} \sum_{i=1}^{m} \left(\boxed{\langle \mathbf{w}' + \mathbf{W}^{\mathrm{src}} \boldsymbol{\beta}, \mathbf{x}_i \rangle} - y_i\right)^2 + \lambda \left\| \mathbf{w}' \right\|_2^2 \right\},$$

(Regularized ERM)

Theorem 1 *Let $h_{\hat{\mathbf{w}},\boldsymbol{\beta}}$ be generated by Regularized ERM, given a m-sized training set S sampled i.i.d. from the target domain, source hypotheses $\{h_i^{src} : \|h_i^{src}\|_\infty \le 1\}_{i=1}^n$, any source weights $\boldsymbol{\beta}$ obeying $\Omega(\boldsymbol{\beta}) \le \rho$, and $\lambda \in \mathbb{R}_+$. Assume that $\ell(h_{\hat{\mathbf{w}},\boldsymbol{\beta}}(\mathbf{x}), y) \le M$ for any $(\mathbf{x}, y)$ and any training set. Then, denoting $\kappa = \frac{H}{\sigma}$ and assuming that $\lambda \le \kappa$, we have with probability at least $1 - e^{-\eta}$, $\forall \eta \ge 0$*

$$R(h_{\hat{\mathbf{w}},\boldsymbol{\beta}}) \le \hat{R}_S(h_{\hat{\mathbf{w}},\boldsymbol{\beta}}) + \mathcal{O}\left(\frac{R^{src}\kappa}{\sqrt{m}\lambda} + \sqrt{\frac{R^{src}\rho\kappa^2}{m\lambda}} + \frac{M\eta}{m\log\left(1+\sqrt{\frac{M\eta}{u^{src}}}\right)}\right) \tag{4}$$

$$\le \hat{R}_S(h_{\hat{\mathbf{w}},\boldsymbol{\beta}}) + \mathcal{O}\left(\frac{\kappa}{\sqrt{m}}\left(\frac{R^{src}}{\lambda} + \sqrt{\frac{R^{src}\rho}{\lambda}}\right) + \frac{\kappa}{m}\left(\frac{\sqrt{R^{src}M\eta}}{\lambda} + \sqrt{\frac{\rho}{\lambda}}\right)\right), \tag{5}$$

where $u^{src} = R^{src}\left(m + \frac{\kappa\sqrt{m}}{\lambda}\right) + \kappa\sqrt{\frac{R^{src}m\rho}{\lambda}}$.

where

$$h_{\boldsymbol{\beta}}^{\mathrm{src}}(\mathbf{x}) := \sum_{i=1}^n \beta_i h_i^{\mathrm{src}}(\mathbf{x}),$$

$$h_{\mathbf{w},\boldsymbol{\beta}}(\mathbf{x}) := \langle \mathbf{w}, \mathbf{x} \rangle + h_{\boldsymbol{\beta}}^{\mathrm{src}}(\mathbf{x})$$

function Ω is σ-strongly convex

$\ell : \mathcal{Y} \times \mathcal{Y} \mapsto \mathbb{R}_+$ is H-*smooth*

$$R^{\mathrm{src}} := R(h_{\boldsymbol{\beta}}^{\mathrm{src}})$$

Remark: the excess risk shrinks at a fast rate of O(1/m). In other words, good prior knowledge guarantees not only good generalization, but also fast recovery of the performance of the best hypothesis in the class.

Learning Categories from Few Examples with Multi Model Knowledge Transfer

Tatiana Tommasi, Francesco Orabona, and Barbara Caputo

- *T. Tommasi is with KU Leuven, ESAT-PSI and iMinds, Leuven 3001, Belgium. E-mail: tatiana.tommasi@esat.kuleuven.be.*
- *F. Orabona is with Toyota Technological Institute at Chicago, Chicago, IL 60637 USA. E-mail: francesco@orabona.com.*
- *B. Caputo is with the University of Rome La Sapienza, Department of Computer, Control and Management Engineering, Rome 00185, Italy. E-mail: caputo@dis.uniroma1.it.*

TPAMI’14

Target model

- Biased Regularized Least Squares SVM

$$\min_{w,b} \ \frac{1}{2}\left\| w - \sum_{j=1}^{J} \beta_j \hat{w}_j \right\|^2 + \frac{C}{2}\sum_{i=1}^{N} \zeta_i \xi_i^2$$

$$\text{s.t.} \quad y_i = w^\top \phi(x_i) + b + \xi_i, \ \forall i = 1, \ldots, N,$$

where

$$\zeta_i = \begin{cases} \frac{N}{2N^+} & \text{if } y_i = +1 \\ \frac{N}{2N^-} & \text{if } y_i = -1 \end{cases}.$$

slack variables ξ_i

(weights of the examples, to balance the different classes)

Source model weights

- The source models are weighted by their LOO error on the labeled target data

Proposition :

the prediction $\tilde{y}_i$, obtained on sample i when it is removed from the training set, is equal to

$$y_i - \frac{a'_i}{P_{ii}} + \frac{\boldsymbol{\beta}^\top \boldsymbol{A}''_i}{P_{ii}}$$

where

P, a', A'' are the quantities that are already computed during the training phase.

Source model weights

$$\ell(\tilde{y}_i, y_i) = \zeta_i |1 - y_i \tilde{y}_i|_+ = \zeta_i \left| y_i \frac{a_i' - \boldsymbol{\beta}^\top A_i''}{P_{ii}} \right|_+ ,$$

where $|x|_+ = \max\{0, x\}$.

$$\min_{\boldsymbol{\beta}} \sum_{i=1}^{N} \ell(y_i, \tilde{y}_i) \quad \text{subject to} \quad \|\boldsymbol{\beta}\|_p \leq 1 \ , \quad \beta_j \geq 0 \ ,$$

- Time complexity $\mathcal{O}(N^3 + JN^2)$

N: The number of training examples

J: The number of source models

Experiments

Dataset: Caltech-256

Setting: leave-one-classout approach, that is considering in turn each class as target and all the others as sources.

Compared methods:

- *Adaptive SVM (A-SVM).* ICDM'07

$$\min_{\boldsymbol{w}} \|\boldsymbol{w} - \beta \hat{\boldsymbol{w}}\|^2 + C \sum_{i=1}^{N} \ell^H(\boldsymbol{w}^\top \phi(\boldsymbol{x}_i), y_i)$$

- *Projective Model Transfer SVM (PMT-SVM).* ICCV'11

$$\min_{\boldsymbol{w}} \ \|\boldsymbol{w}\|^2 + \beta \|\mathbf{R}\boldsymbol{w}\|^2 + C \sum_{i=1}^{N} \ell^H(\boldsymbol{w}^\top \phi(\boldsymbol{x}_i), y_i)$$

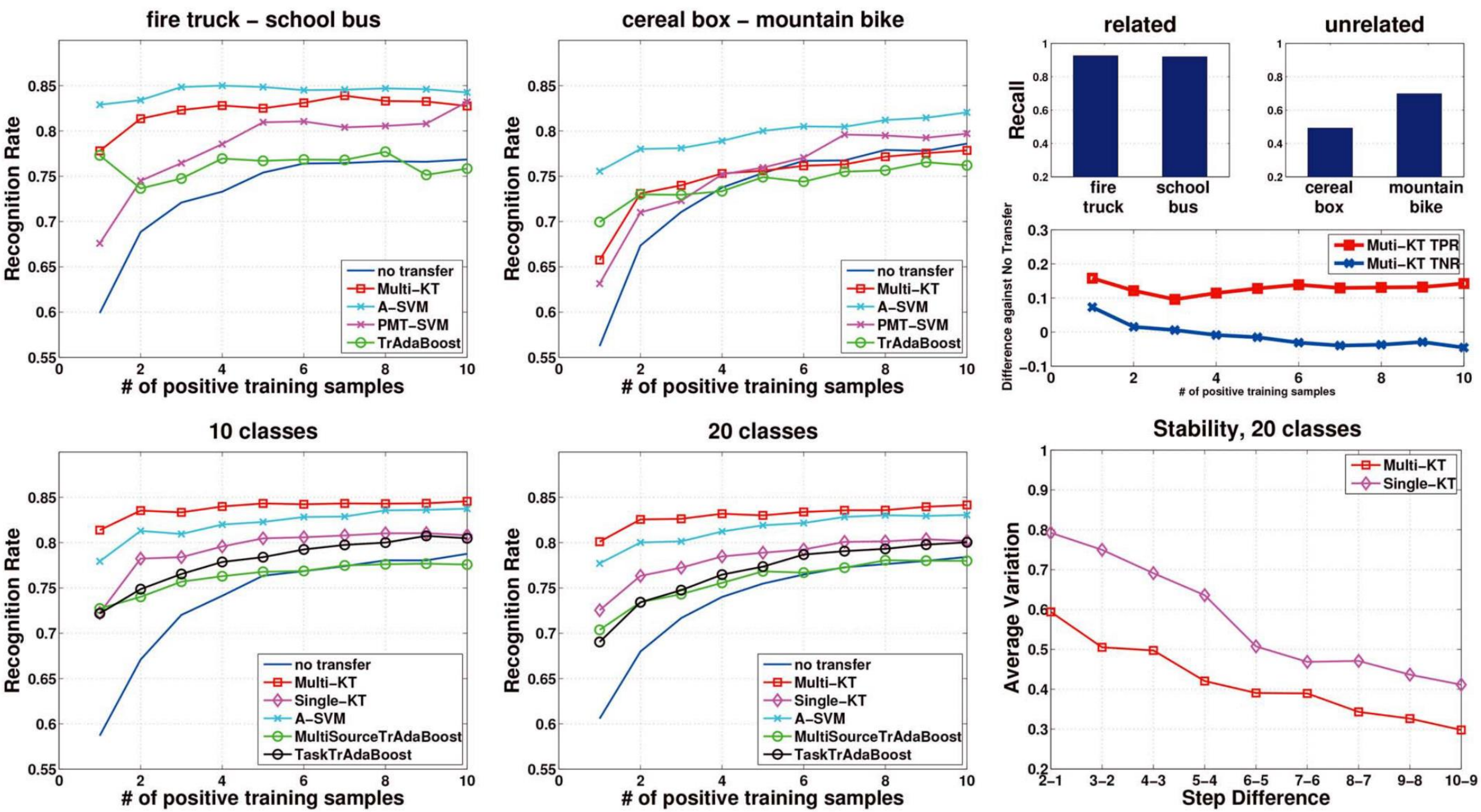
$$\text{s. t.} \quad \boldsymbol{w}^\top \hat{\boldsymbol{w}} \geq 0, \quad \|\mathbf{R}\boldsymbol{w}\|^2 = \|\boldsymbol{w}\|^2 \sin^2 \theta,$$

where θ is the angle between $\boldsymbol{w}$ and $\hat{\boldsymbol{w}}$.

- *TrAdaBoost: boosting for Transfer Learning* ICML'07

  starting from the combination of source and target samples, iteratively decreases the weights of the source data in order to weaken their impact on the learning process

Main results



模式识别与神经计算研究组

PAttern Recognition and NEural Computing

Handling Concept Drift via Model Reuse

Peng Zhao[1] · Le-Wen Cai[1] · Zhi-Hua Zhou[1]

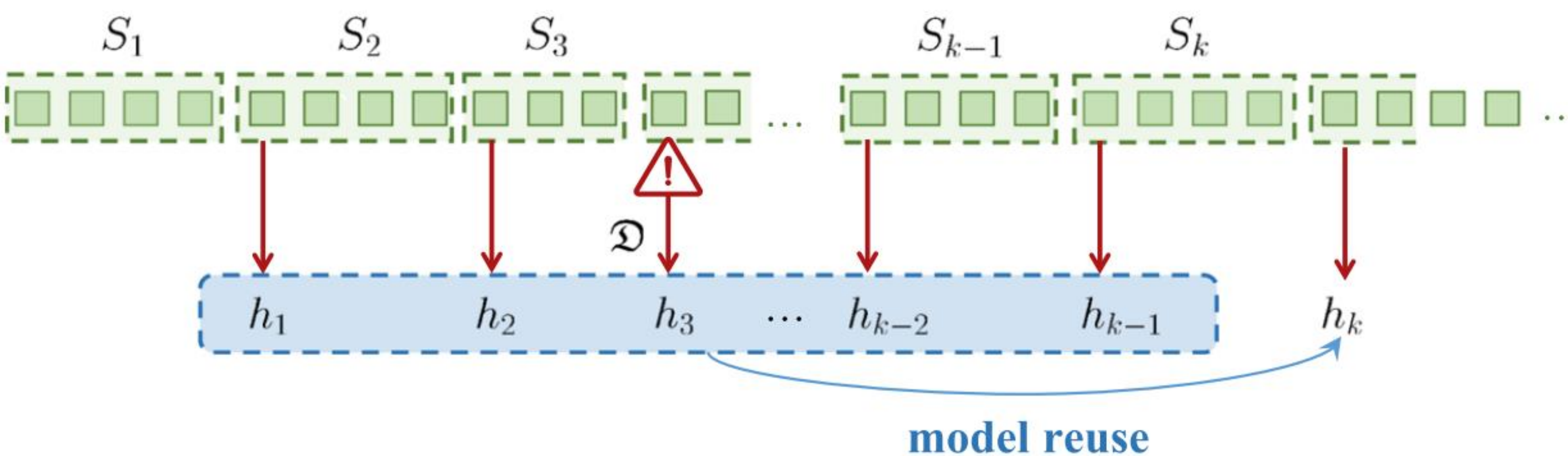
1 National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China

ML'20

The method

- **<u>Setting</u>**: data are coming one by one sequentially, and there may emerge concept drift in the data stream.

- **<u>Framework</u>**: when the maximum update period is achieved, or the abrupt change is detected, train a new model with the help with historical models.



Target model

- Biased Regularized Least Squares SVM

$$\hat{\mathbf{w}}_k = \arg\min_{\mathbf{w}} \left\{ \frac{1}{m} \sum_{i=1}^{m} \ell\left(\langle \mathbf{w}, \mathbf{x}_i \rangle, y_i \right) + \mu \Omega\left(\mathbf{w}, \mathbf{w}_p \right) \right\},$$

where

$$\Omega\left(\mathbf{w}, \mathbf{w}_p\right) = \|\mathbf{w} - \mathbf{w}_p\|^2$$

$$\mathbf{w}_p = \sum_{j=1}^{k-1} \beta_j \hat{\mathbf{w}}_j$$

Source model weights

- The historical models are weighted by their performance on the labeled target data
  - Weight update by expert advice

$$\beta_{t+1,k} \propto \beta_{t,k} \exp\{-\eta \ell(\hat{y}_{t,k}, y_t)\}.$$

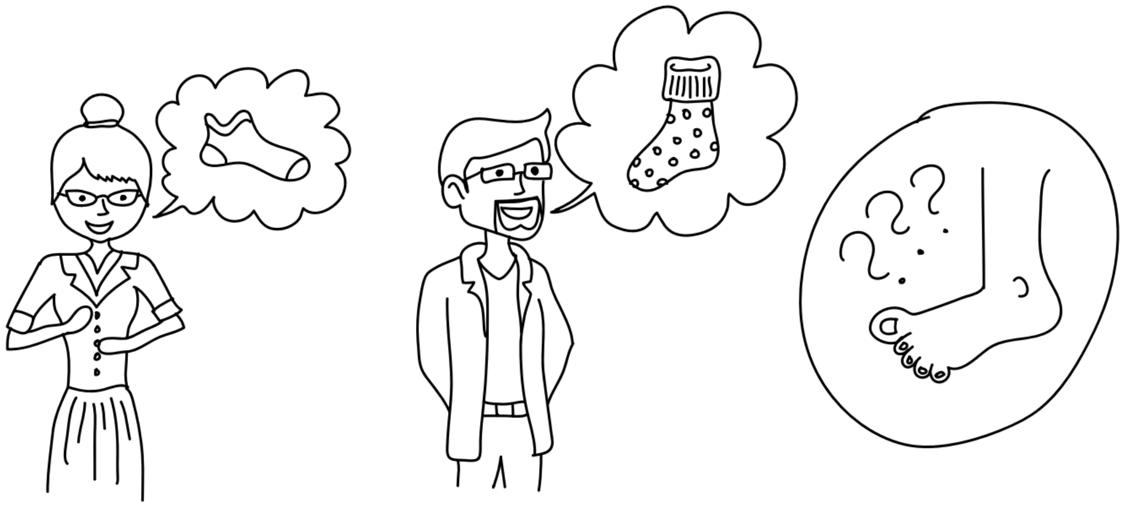


Figure 18.2 Prediction with expert advice. The experts, upon seeing a foot give expert advice on what socks should fit it best. If the owner of the foot is happy, the recommendation system earns a cookie!

Experiments

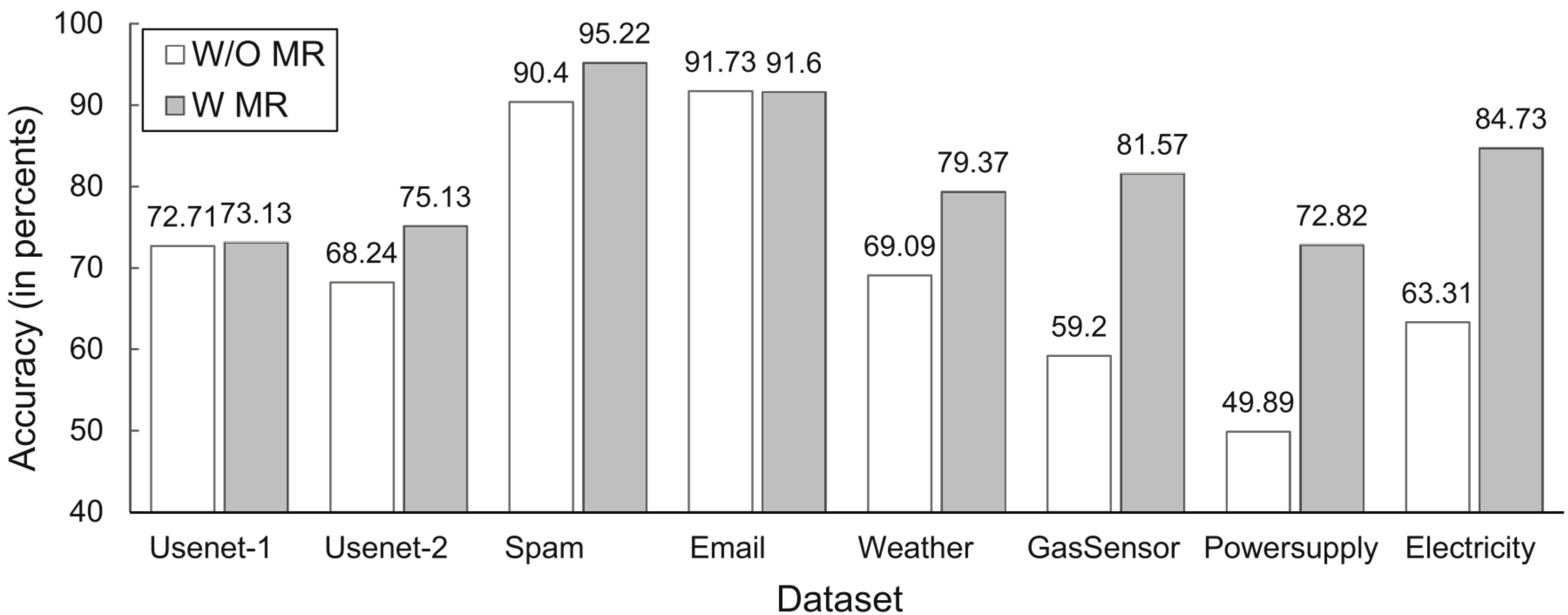


Fig. 3 Performance comparisons (in predictive accuracy) of Condor with/without model reuse mechanism

Contents

ParNeC 模式识别与神经计算研究组 PAttern Recognition and NEural Computing

模式识别与神经计算研究组

PAttern Recognition and NEural Computing

Model Reuse with Reduced Kernel Mean Embedding Specification

Xi-Zhu Wu[1], Wenkai Xu[2], Song Liu[3,4], Zhi-Hua Zhou[1]*

[1]National Key Laboratory for Novel Software Technology, Nanjing University, China.

[2]Gatsby Unit of Computational Neuroscience, University College London, UK.

[3]School of Mathematics, University of Bristol, UK. [4]The Alan Turing Institute, UK.

Arxiv'20

- Summary

  Using the distribution difference to weigh the source models.
  Requires to compute and upload the RKME values to represent the distributions of source data (will not expose the raw data).

- Kernel Mean Embeddings （KME）

$$\mu_k(P) := \int_{\mathcal{X}} k(x, \cdot)\, \mathrm{d}P(x) \qquad k(x, x') = \exp(-\gamma\|x - x'\|), \gamma > 0.$$

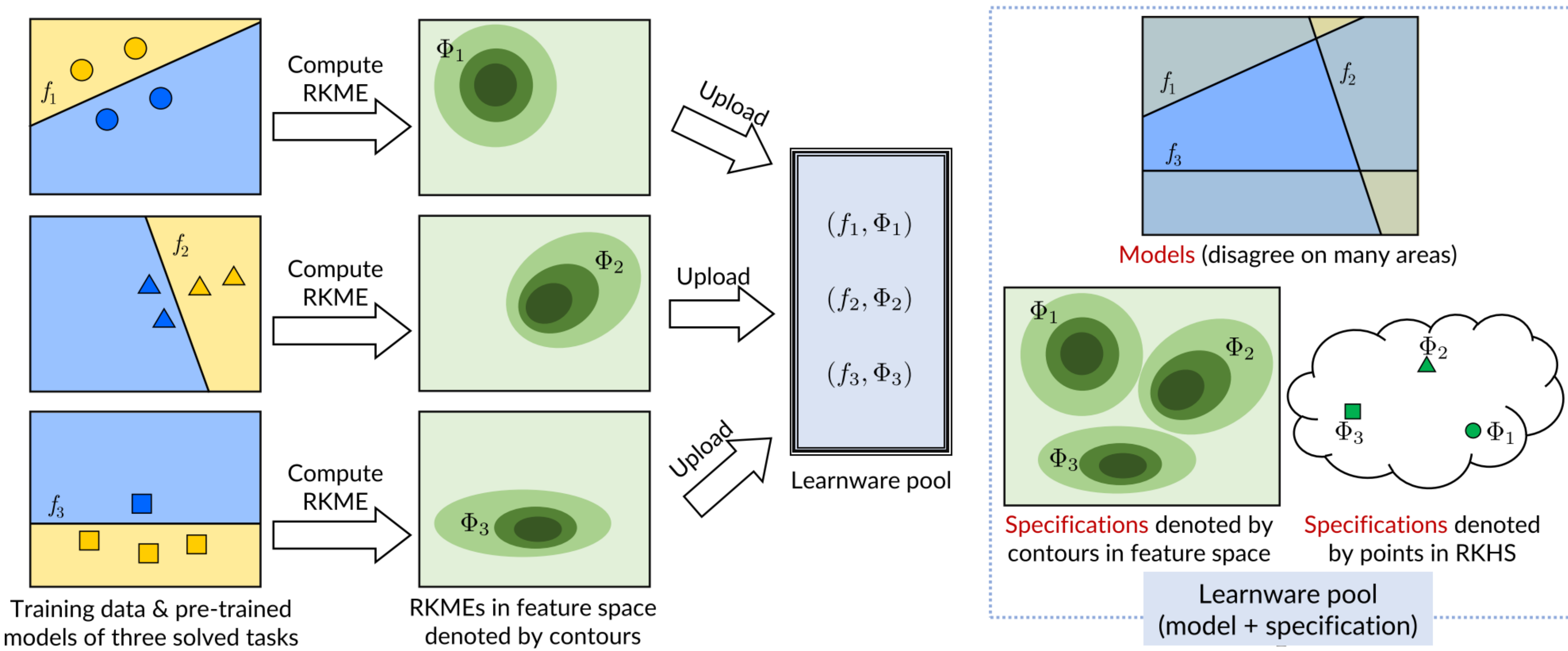
$$\widehat{\mu}_k(P_X) := \frac{1}{N}\sum_{n=1}^{N} k(x_n, \cdot) \qquad X \in \mathcal{X} \qquad X = \{x_n\}_{n=1}^{N} \sim P^N,$$

- Reduced Set Construction

  $\{z_m\}$ are newly constructed vectors

$$\|\widehat{\mu}_k(P_X) - \widehat{\mu}_k(P_R)\|^2_{\mathcal{H}_k} = \left\| \sum_{n=1}^{N} \frac{1}{N} k(x_n, \cdot) - \sum_{m=1}^{M} \beta_m k(z_m, \cdot) \right\|^2_{\mathcal{H}_k}$$

Upload phase



Φ_i (represented by β and Z),

Figure 1: An illustration of the upload phase.

Deployment phase

- Task-recurrent assumption

The target task has the same distribution with one of the source task.

Solution:

if task-recurrent assumption **then**

$$\Phi_t = \sum_{n=1}^{N} \frac{1}{N} k(x_n, \cdot)$$

$$i^* = \arg\min_i \|\Phi_t - \Phi_i\|^2_{\mathcal{H}_k}$$

$$Y = \widehat{f}_{i^*}(X)$$

end if

- Instance-recurrent assumption

The distribution of the current task is a mixture of solved tasks.

Solution: "recover" enough data points from test distribution and learn a model selector on them.

Deployment phase

$$\min_{w} \left\| \frac{1}{N} \sum_{n=1}^{N} k(x_n, \cdot) - \sum_{i=1}^{c} w_i \Phi_i(\cdot) \right\|_{\mathcal{H}_k}^2, \qquad (12)$$

$$x_{T+1} = \begin{cases} \arg\max_{x \in \mathcal{X}} \Phi(x), & \text{if } T = 0 \\ \arg\max_{x \in \mathcal{X}} \Phi(x) - \frac{1}{T+1} \sum_{t=1}^{T} k(x_t, x), & \text{if } T \geq 1. \end{cases} \qquad (13)$$

Estimate $\widehat{w}$ as (12)
Initialize the mimic sample set $S = \emptyset$
while $|S|$ is not big enough **do**
  Sample a provider index i by weight $\widehat{w}_i$
  Sample an example x by kernel herding as (13)
  $S = S \cup \{(x, i)\}$
end while
Train a selector g on mimic sample S
for $n = 1 : N$ **do**
  $i^* = g(x_n)$
  $y_n = \widehat{f}_{i^*}(x_n)$
end for

Deployment phase

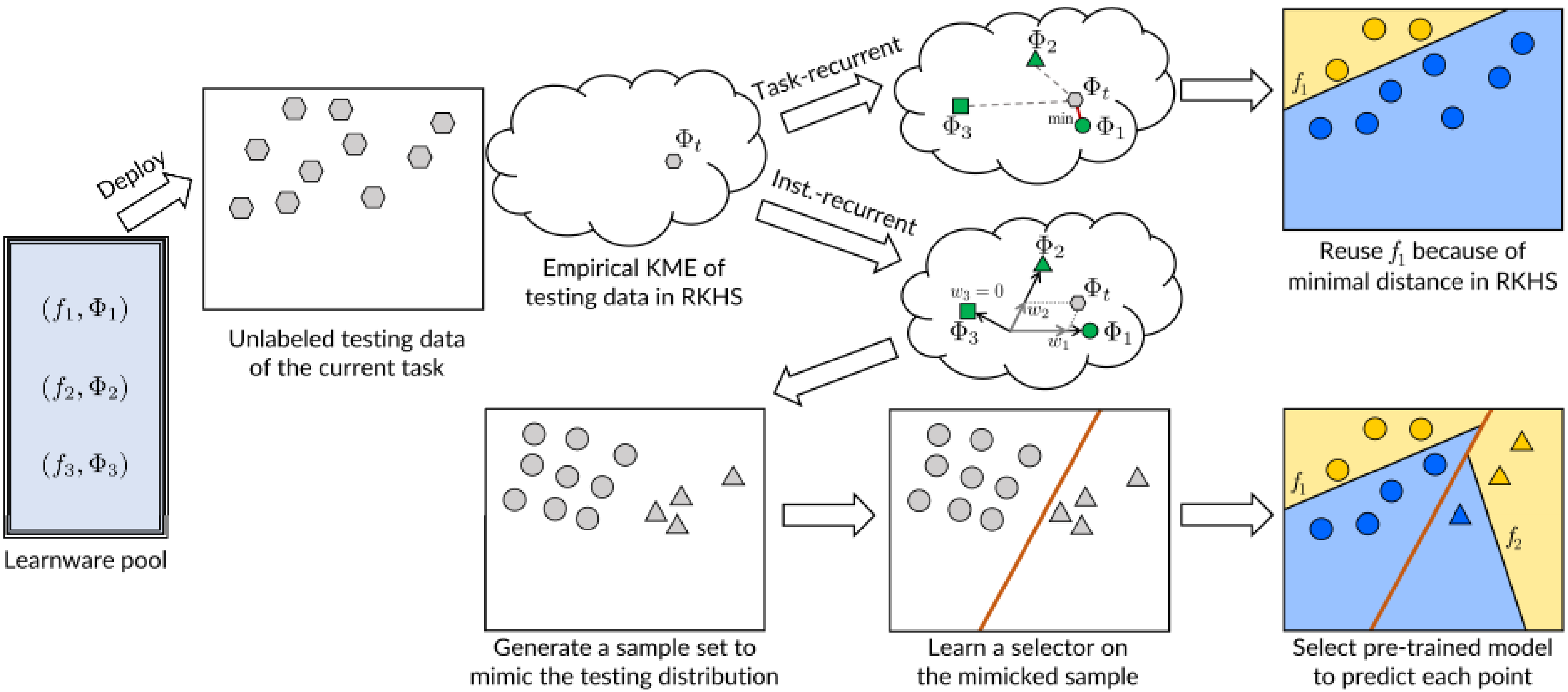


Figure 2: An illustration of the deployment phase.

Experiments

Dataset: CIFAR-100, 20-newsgroup

Setting:

- Divide CIFAR-100 **into 20 local datasets**, each having images from one superclass, and **build 5-class local neural network classifiers on them**.

- For 20-newsgroup, there are **5 superclasses** {comp, rec, sci, talk, misc} and **each is considered a local dataset** for training local models in the upload phase.

Compared methods:

- MAX

  simply uses all the pre-trained models to predict one test instance, and takes out the most confident predicted class.

- HMR (ICML19)

  HMR incorporates a communication protocol which exchanges several selected key examples to update models.

Table 1: Results of CIFAR-100 in accuracy(%).

	Task-recurrent	Instance-recurrent			
#Mixing tasks	1	2	5	10	20
MAX	43.00	42.10	41.51	41.62	41.44
HMR	70.58	68.91	68.93	68.88	**68.81**
Ours	**86.22**	**72.91**	**72.57**	**71.07**	68.79
Global	75.08	73.24	73.31	71.86	73.24

a global model trained on merged data.

Table 2: Results of 20-newsgroup in accuracy(%).

	Task-recurrent	Instance-recurrent			
#Mixing tasks	1	2	3	4	5
MAX	58.65	55.76	53.03	51.94	50.68
HMR	72.01	72.19	70.86	70.53	70.09
Ours	**83.13**	**76.03**	**75.10**	**74.02**	**72.68**
Global	72.06	73.24	73.31	71.86	73.24

Inductive Model Reuse via Synergistic Training

Anonymous

Submit to ICML'21

Problem setting

- Source tasks may have different label spaces. But all the tasks are sampled from the common space $\mathcal{X}, \mathcal{Y}$

- Train one-class classifiers for each class. (for all source models)

- Properly select M_l source models for prediction, rather than training a new one.

$$\bar{f}_l(x) = \frac{1}{M_l} \sum_{m=1}^{M_l} f_{l_{k_m}, l_{j_m}}(x).$$

Heuristic criteria

- Class-wise margin:

$$\frac{1}{N}\sum_{n=1}^{N}\left(f_{k,j}(x_{n,l}) - \max_{j'\neq j} f_{k,j'}(x_{n,l})\right).$$

- Instance-wise margin:

$$\frac{1}{N}\sum_{n=1}^{N}\left(f_{k,j}(x_{n,l}) - \max_{l'\neq l} f_{k,j}(x_{n,l'})\right).$$

where $x_{n,l}$ n-th instance of the l-th class

$f_{k,j}$ k-th source task of j-th class

Learned criteria

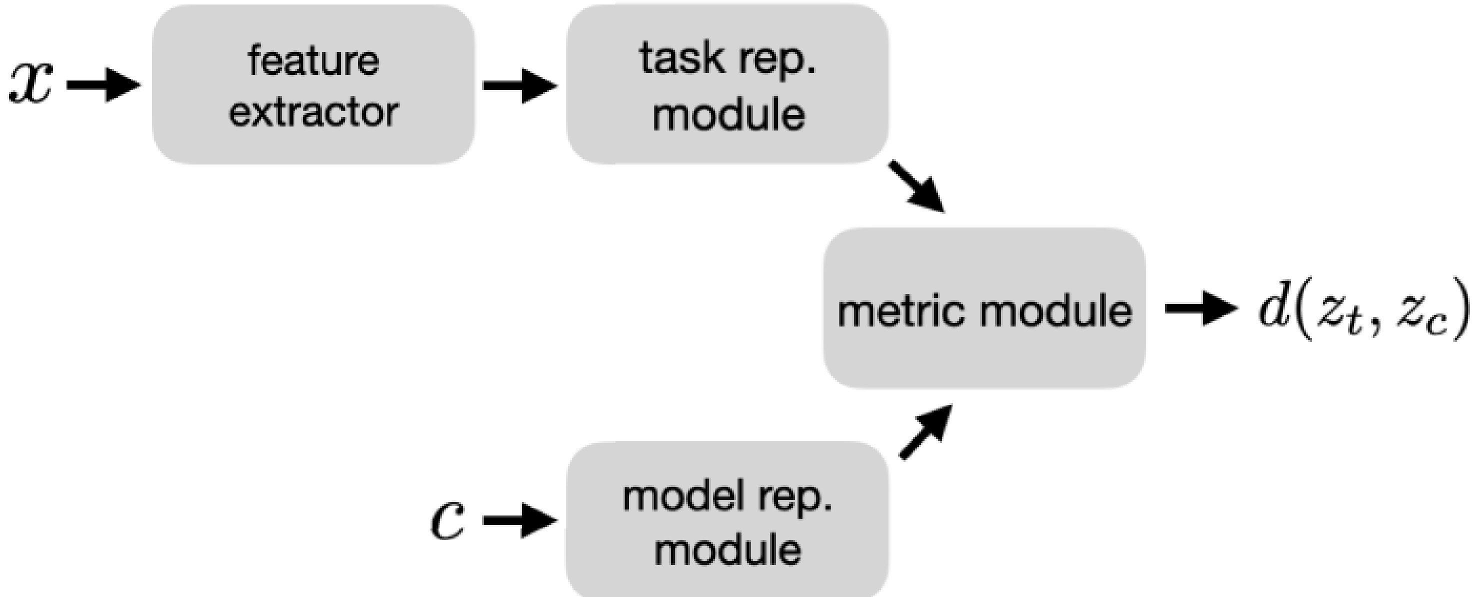


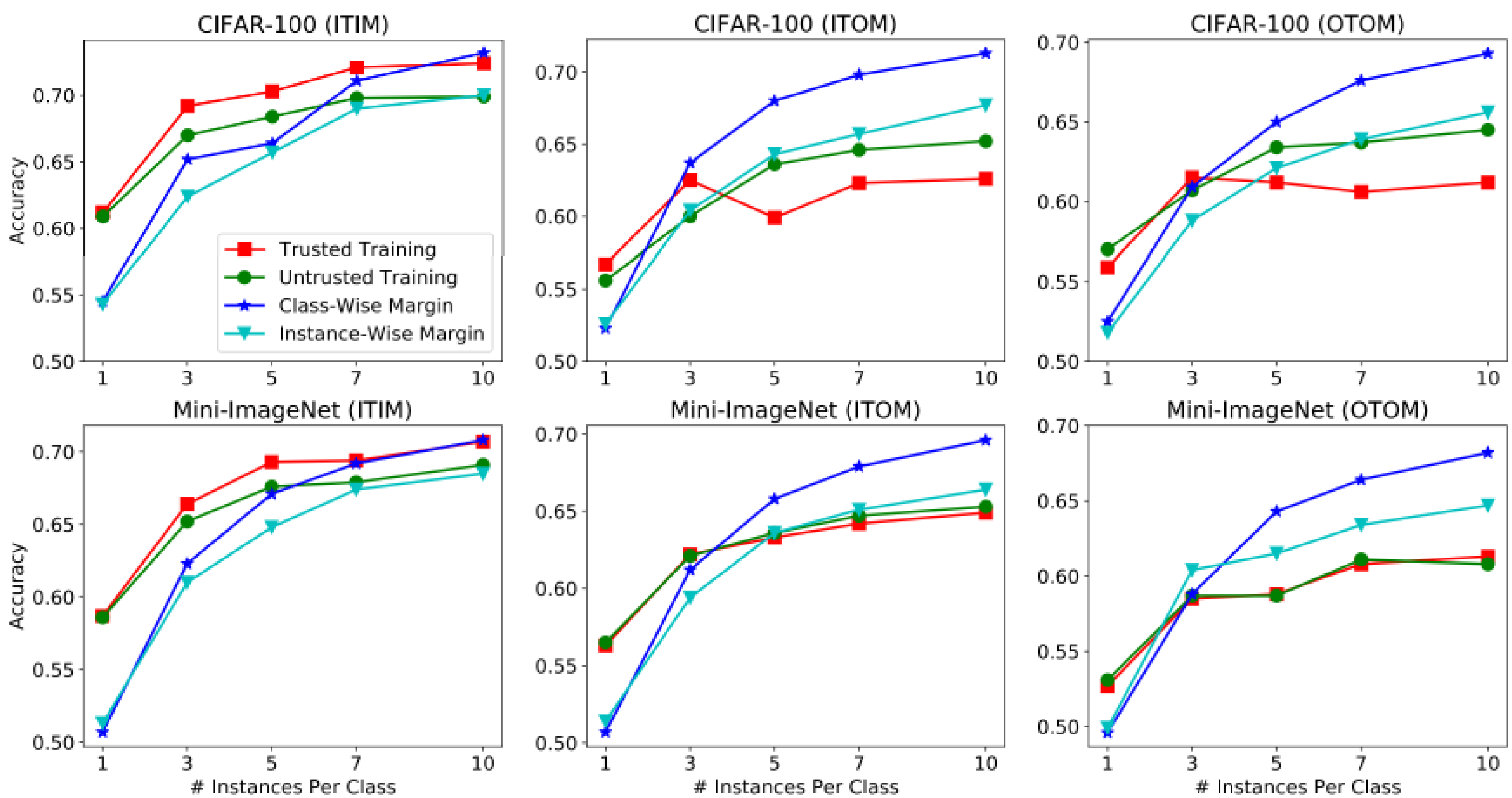
Figure 1. The structure of MoreNet .

If a model c is reusable on task t, the distance between their embeddings should be small.

Experiments

Table 1. The averaged accuracy on benchmark datasets. The validation and test margins indicate the class-wise margin based result over validation set of size 100 and test data directly. All results are average accuracy (%)± standard deviation (%).

	F-MNIST	CIFAR-10	CIFAR-100 (ITIM)	CIFAR-100 (ITOM)	CIFAR-100 (OTOM)	Mini-ImageNet (ITIM)	Mini-ImageNet (ITOM)	Mini-ImageNet (OTOM)
Class-Wise Margin	87.62±2.58	68.70±3.46	54.51±2.97	52.33±2.93	52.51±1.81	50.70±3.67	50.68±4.22	49.58±2.32
Instance-Wise Margin	87.71±2.82	70.01±3.15	54.29±2.76	52.62±3.59	51.83±2.29	51.34±3.40	51.39±3.90	49.86±1.98
Trusted Training	**89.64±2.55**	**76.59±1.96**	60.87 ±3.98	55.58±3.47	55.90±1.73	**58.73±3.12**	56.34±3.41	52.74±2.88
Untrusted Training	83.68±3.49	74.25±2.14	**61.19±4.19**	**56.69±3.66**	**57.02±2.06**	58.62±3.67	**56.47±3.34**	**53.06±2.64**
Validation Margin	93.32±1.21	83.00±2.41	74.66±3.40	72.30±3.15	72.05±3.22	72.90±3.05	71.73±2.01	69.71±3.05
Test Margin	93.65±1.54	83.54±2.25	82.86±1.30	80.27±2.94	79.49±2.13	80.56±1.78	78.61±1.75	77.74±1.48



In-Task-In-Model (ITIM) : both the testing-stage label space and the models are the same to the training stage

Conclusion

- Model reuse methods use different criteria to evaluate the reusability of source models on the target tasks. E.g., exploit the labeled target data, compare the distribution gap, etc.

- Many of the existing work learn a target model with the biased regularization due to its theoretical properties. While the others try to select eligible source models to predict the target data.

Discussion

ParNeC 模式识别与神经计算研究组 PAttern Recognition and NEural Computing

- Extend the model reuse framework to semi-supervised learning

$$\boldsymbol{w}_S = \operatorname*{argmin}_{\boldsymbol{u}} \frac{1}{m} \sum_{i=1}^{m} (\boldsymbol{u}^\top \boldsymbol{x}_i - y_i)^2 + \lambda \|\boldsymbol{u} - \boldsymbol{w}'\|^2 + SSL\_regularizer$$

- Extend the model reuse to active learning.

Motivation: When the initial labeled data is limited, the target model is unreliable. Incorporate the source models may reduce the model uncertainty, and lead to better data selection & model learning.

Shall we design an unified objective for target model learning, source model reweighting and data selection?