# Towards Enabling Learnware to Handle Unseen Jobs

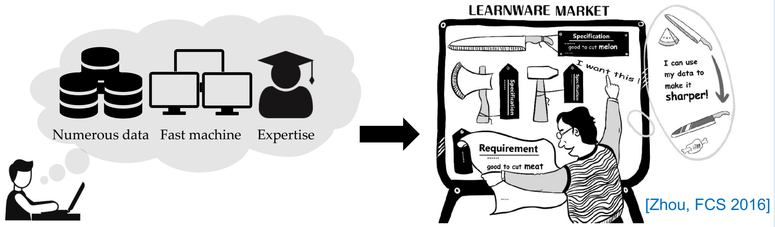
Yu-Jie Zhang, Yu-Hu Yan, Peng Zhao, Zhi-Hua Zhou

**Contact**

{zhangyj, yanyh, zhaop, zhouzh}
@lamda.nju.edu.cn

## Learnware Paradigm

The **learnware paradigm** (Zhou 2016) attempts to change the current style of machine learning deployment:



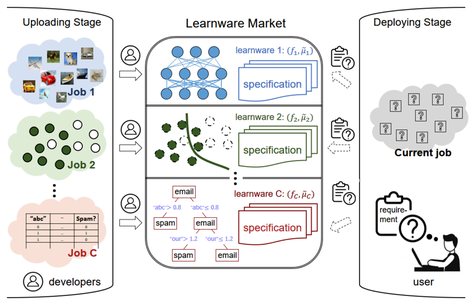

[Zhou, FCS 2016]

Current style: Learn from scratch

Learnware paradigm: Reuse previous efforts

**A Learnware** = A well-performed pre-trained **model** + **specification** explaining its purpose and/or specialty

Sharing the previous efforts by maintaining a platform containing various learnware primarily developed for different jobs.




How to design specification?

- **Reduced Kernel Mean Embedding [Wu et al. 2020]:** approximate the original data by a small set of weighted samples.

$$\min_{\beta, Z} \left\| \sum_{n=1}^{N} \frac{1}{N} k(x_n, \cdot) - \underbrace{\sum_{m=1}^{M} \beta_m k(z_m, \cdot)}_{\text{RKME: } \tilde{\mu}_i} \right\|_{\mathcal{H}}^2$$

KME of original data

**RKME** is a nice specification
- Capture sufficient information
- Protect data privacy

*Our main focus: How to reuse models?*

- Challenge: Potentially useful learnware could be very few.
- One should identify **which** models to reuse, and know **how** to reuse them.

## Handling Unseen Jobs

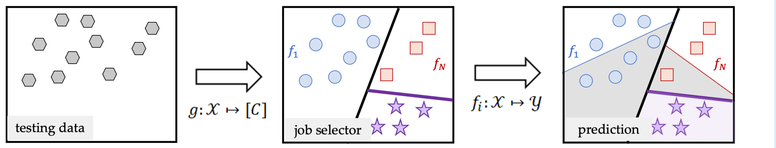
Previous studies: user's job is well covered by learnware market.

**This work:** consider the existence of **unseen parts** in user's job.

❖ **Unseen-job assumption:** The testing distribution $\mathcal{D}_{\mathrm{XY}}^{\mathrm{te}}$ is a **mixture** of those of the uploaded jobs $\mathcal{D}_{\mathrm{XY}}^{i}$ and an unseen job $\mathcal{D}_{\mathrm{XY}}^{\mathrm{u}}$, as $\mathcal{D}_{XY}^{te} = \sum_{i=1}^{C} w_i \mathcal{D}_{XY}^{i} + w_{\mathrm{u}} \mathcal{D}_{XY}^{\mathrm{u}}$.

## Our Approach

**Job Selector:** train a classifier $g: \mathcal{X} \mapsto \{1, \ldots, C, \mathrm{u}\}$ to assign each sample a job and we make the final prediction by $f(\mathbf{x}) = f_{g(\mathbf{x})}(\mathbf{x})$.




The classification error of final prediction can be decomposed as

$$\mathbb{E}_{\mathcal{D}_{XY}^{te}} \left[ L_{01}(f(\mathbf{x}), y) \right] \leq \underbrace{\sum_{i=1}^{C} \mathbb{E}_{\mathcal{D}_{XY}^{i}} \left[ \mathbb{1}\left(f_i(\mathbf{x}) \neq y\right) \right]}_{\text{error of the pre-trained model}} + R(g),$$

The pre-trained models can perform well on their own job, we only require to minimize $R(g)$, the error of the job selector.

**Risk Decomposition:** the job selector's error has three parts

$$R(g) = \sum_{i=1}^{C} w_i \mathbb{E}_{\mathcal{D}_X^i} \left[ L_{01}(g(\mathbf{x}), i) \right] + w_{\mathrm{u}} \mathbb{E}_{\mathcal{D}_X^u} \left[ L_{01}(g(\mathbf{x}), \mathrm{u}) \right]$$

How to assess their values ?

❖ Term A: error of the job selector over the $i$-th job.

Intuition: $\tilde{\mu}_i$ can be used to approximate $\mathcal{D}_{\mathrm{X}}^{i}$.
Technique: **kernel herding**.

❖ Term B: $w_i, w_{\mathrm{u}}$, the proportion of jobs in the testing data.

Intuition: the proportion of $\tilde{\mu}_i$ in $\mu_{te}$ can be estimated.
Technique: **mixture proportion estimation**.

❖ Term C: error of the job selector over the unseen job.

Intuition: dist. of the unseen job hides in that of the testing data.
Technique: **expected risk rewriting**.

## Theoretical Analysis

**Theorem 1.** *Let $k(\mathbf{x}, \mathbf{x}) \leq 1$ for all $\mathbf{x} \in \mathcal{X}$ and the binary loss function $\psi$ is bounded by $B_\psi \geq 0$ and is L-Lipschitz continuous.[2] Then, for all $g_1, \ldots, g_C, g_u \in \mathcal{G}$. We have*

$$R_\Psi(\widetilde{\mathbf{g}}) - R_\Psi(\mathbf{g}^*) \leq O\left( \frac{C}{\sqrt{N}} + \frac{C}{\sqrt{N_t}} + \frac{1}{\sqrt{M}} \right).$$

*where $\mathbf{g}^* = \arg\min_{g_1, \ldots, g_C, g_u \in \mathcal{G}} R_\Psi(\mathbf{g})$ is the optimal classifier minimizing the expected risk $R_\Psi$ over $\mathcal{G}$.*

**Excess Risk Bound:** our job selector converges to the optimal one.

**Theorem 2.** *Let $k(\mathbf{x}, \mathbf{x}) \leq 1$ for all $\mathbf{x} \in \mathcal{X}$. Let the kernel $k$, and distribution $\mathcal{D}_X^i$, $\bar{P}_X^{(i)}$ satisfy the separability condition with tolerance $\beta$ and margin $\alpha > 0$. Let $\nu \in \left[ \frac{\alpha}{4\lambda_i}, \frac{3\alpha}{4\lambda_i} \right]$, $\sqrt{\min\{N, M, N_t\}} \geq \frac{48\sqrt{\log(1/\delta)}}{\alpha/\lambda_i - \nu}$ and $\widetilde{\mu}_i \in \widehat{\mathcal{C}}$. We have*

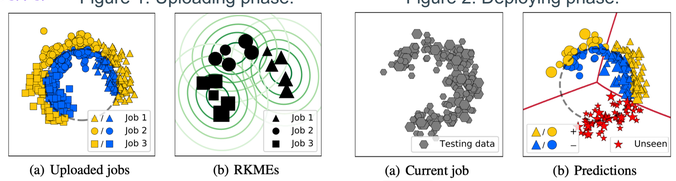
$$\lambda_i - \widehat{\lambda}_i^G \leq O\left( \frac{1}{\sqrt{\min\{N, M, N_t\}}} \right),$$

$$\widehat{\lambda}_i^G - \lambda_i \leq 8\beta\lambda_i/\alpha + O\left( \frac{1}{\sqrt{\min\{N, M, N_t\}}} \right).$$

**Convergence:** estimation of weights converges to the true values.

## Experiments

**Toy data:**




**Benchmark data:**

Table 1: Accuracy on true labels.

| Datasets | Job number | Instance-recurrent assumption (2) | | Unseen-job assumption (3) | | | | |
|---|---|---|---|---|---|---|---|---|
| | | RKME-basic | Ours | RKME-OCSVM | RKME-iForest | Ours | Ours-oracle | Oracle |
| CIFAR-100 | 2 | 75.90 ± 5.41 | **79.28 ± 4.84** | 62.81 ± 6.13 | 58.55 ± 4.74 | **90.32 ± 2.02** | 90.39 ± 2.22 | 93.55 ± 1.52 |
| | 5 | **75.43 ± 3.92** | 72.59 ± 4.80 | 49.80 ± 2.73 | 37.85 ± 2.64 | **76.04 ± 4.84** | 79.15 ± 2.01 | 87.88 ± 2.77 |
| | 10 | **75.95 ± 2.34** | 73.28 ± 2.10 | 44.70 ± 2.02 | 29.18 ± 2.97 | **72.42 ± 3.81** | 74.49 ± 2.71 | 86.43 ± 1.95 |
| Newsgroup20 | 2 | 85.75 ± 7.59 | 84.31 ± 8.34 | 56.44 ± 7.03 | 59.01 ± 5.46 | **79.91 ± 9.98** | 88.73 ± 7.48 | 93.64 ± 4.94 |
| | 3 | 88.18 ± 6.69 | 86.70 ± 6.79 | 52.93 ± 3.81 | 48.58 ± 3.59 | **75.56 ± 6.39** | 83.43 ± 4.21 | 90.55 ± 3.61 |
| | 4 | **87.10 ± 6.38** | 83.32 ± 6.89 | 48.94 ± 2.69 | 43.75 ± 3.53 | **71.19 ± 6.15** | 80.90 ± 2.50 | 89.88 ± 2.09 |
| ELT Character | 2 | 87.47 ± 5.04 | 88.15 ± 4.26 | 31.56 ± 7.56 | 38.15 ± 12.8 | **91.85 ± 2.63** | 93.14 ± 2.06 | 95.67 ± 2.74 |
| | 3 | **87.09 ± 2.35** | 84.15 ± 3.28 | 37.79 ± 9.05 | 37.27 ± 10.0 | **85.52 ± 5.30** | 88.85 ± 4.40 | 95.49 ± 1.61 |
| | 4 | **86.23 ± 2.63** | 81.45 ± 2.93 | 44.75 ± 4.59 | 39.49 ± 6.55 | **76.01 ± 5.53** | 84.61 ± 2.97 | 94.10 ± 1.67 |

Ours

Baselines

Skylines

**Results:**

❖ Our method achieves **comparable** performance with RKME-basic under *instance-recurrent assumption* and outperforms **all** the contenders under *unseen-job assumption*.

❖ The gap between ours and ours-oracle and the gap between ours-oracle and oracle also validate the efficacy of our method.