Identifying Helpful Learnwares without Examining the Whole Market

Yi Xie, Zhi-Hao Tan, Yuan Jiang, Zhi-Hua Zhou

{xiey, tanzh, jiangy, zhouzh} @lamda.nju.edu.cn

http://www.lamda.nju.edu.cn

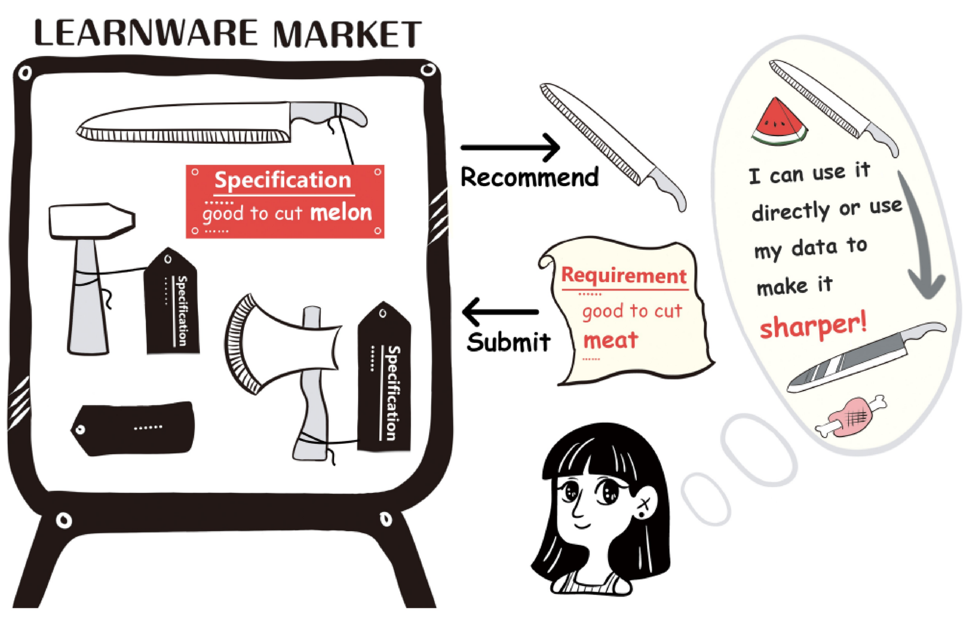
1. Background: Learnware Paradigm

Building high-quality models:

- Complex, time-consuming and expensive: data, computing resources, expertise…
  - ➢ A heavy burden for ordinary users.
- Difficult to reuse among different users: data privacy concerns, catastrophic forgetting.

Learnware paradigm [Zhou, 2016; Zhou and Tan, 2022]

- Construct a **model market** that manages numerous well-performing models.
- Solve future tasks by leverage these models without having to build models from scratch.



1) Learnware components

learnware = model + **specification**

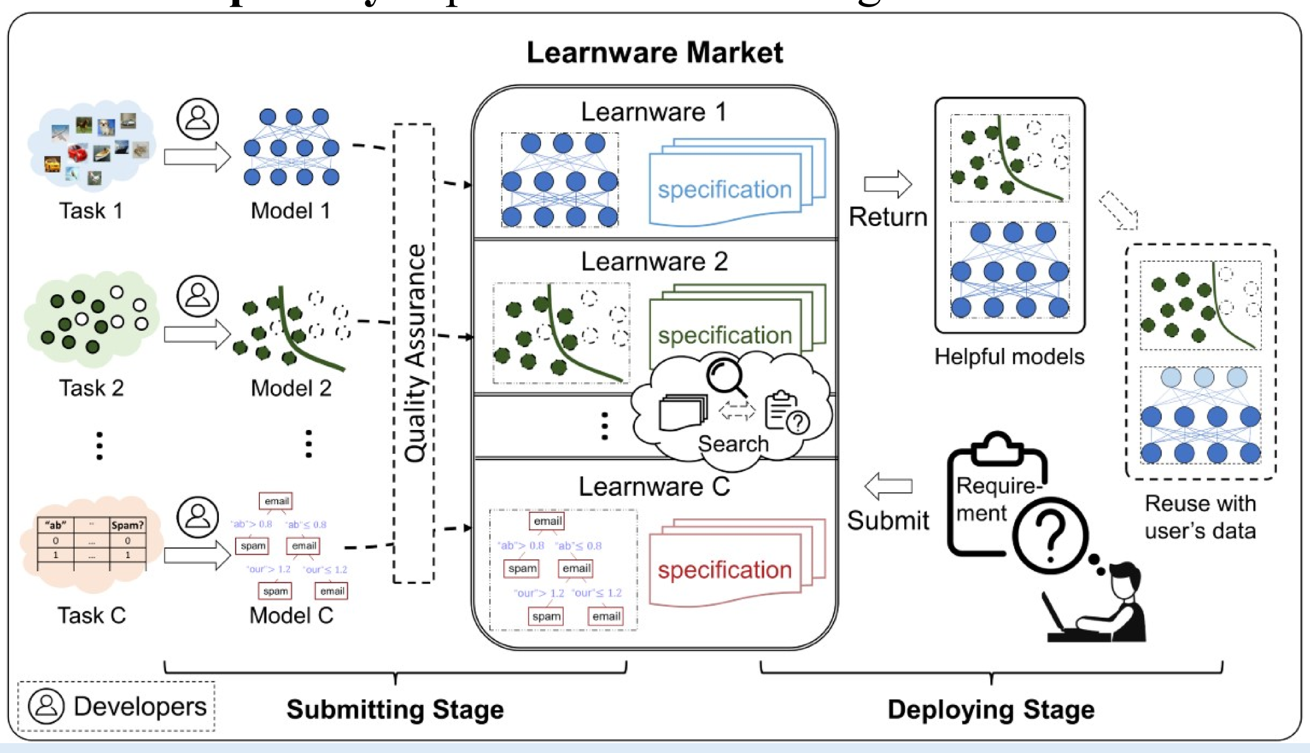
describe the functionality of the model

2) Procedure of learnware paradigm

Submitting stage: The learnware market **assigns specifications** to submitted models.

Deploying stage: The market helps the user **identify & reuse helpful models** according to the requirement.

Data privacy is preserved in both stages.



Key challenge: how to *identify* helpful models for a specific user task *efficiently* without leaking user *data privacy*?

3) Reduced Kernel Mean Embedding (RKME) specification [Zhou and Tan, 2022]

$$\min_{\beta, Z} \left\| \sum_{i=1}^{n} \frac{1}{n} k(x_i, \cdot) - \sum_{j=1}^{m} \beta_j k(z_j, \cdot) \right\|_{\mathcal{H}}^2$$

KME of original data — RKME specification

- Sketch the dataset via weighted samples in RKHS.
- Capture major distribution information while protecting data privacy.
- Assumption: each learnware is a well-performing model on its on training data.
  - ➢ Identifying a suitable model for user task can be approached by identifying a model whose training distribution is close to the distribution of user task.

2. Motivation

Previous algorithms based on RKME specification

[Wu *et al.*, 2023][Zhang *et al.*, 2021] [Tan *et al.*, 2022] [Tan *et al.*, 2023]

- Require examining all learnwares in the market:
  - ➢Computationally unaffordable in large markets.
- Impose strict restrictions on the market:
  - ➢E.g., all learnwares share the same ground-truth labeling function.

This paper: a more efficient and flexible method

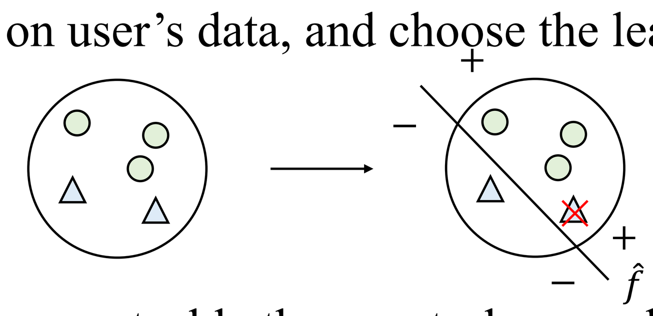
- A learnware scoring criterion with fewer restrictions.
- An anchor-based framework only examining only a small portion of the market.

3. Whether a learnware is helpful?

Question 1: How to judge whether a learnware is potentially helpful based on the limited labeled data of the user?

Case 1: There exists one learnware that can solve user's task.

Solution: Calculate losses on user's data, and choose the learnware with the smallest loss.



Case 2: No single learnware can tackle the user task as a whole, but multiple learnwares can each tackle a part of user's task separately.

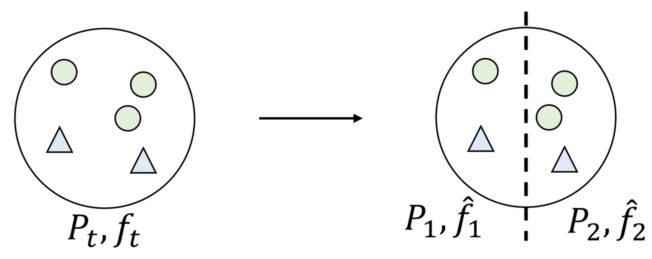
Instance-recurrent Assumptions

- The user's distribution is a mixture of multiple key learnwares' distributions

$$P_t = \sum_i w_i P_i$$

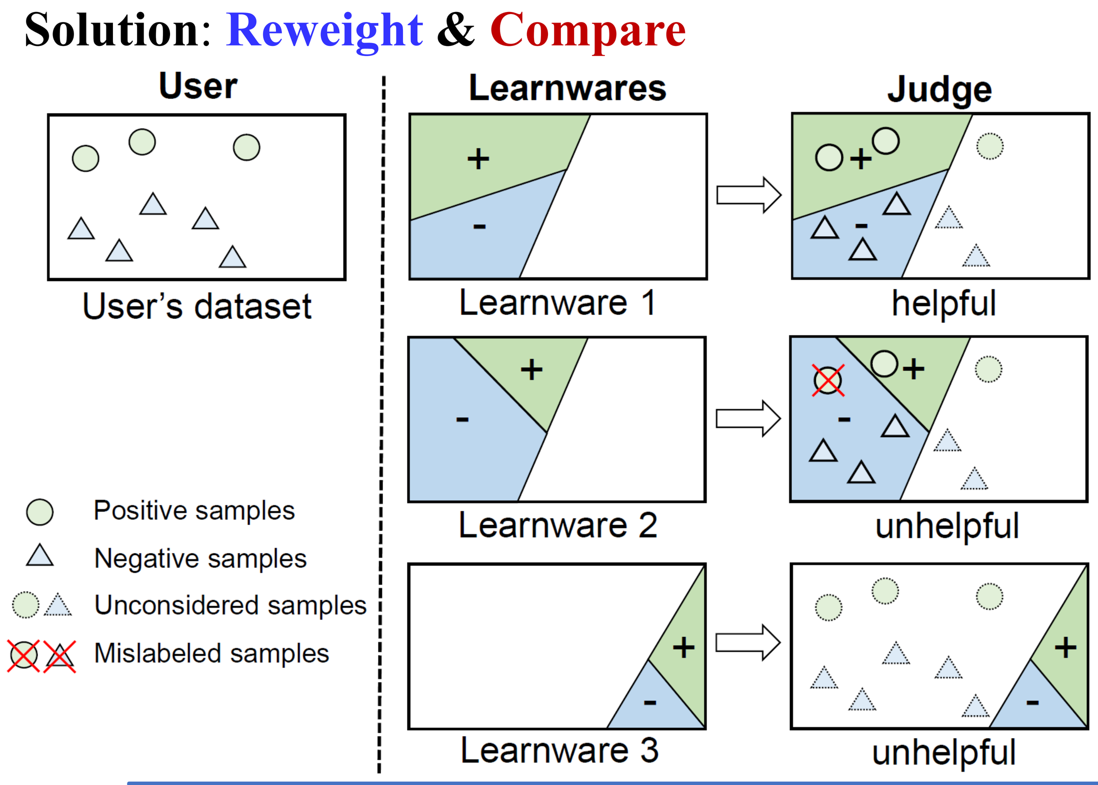
- Each key learnware i performs well in the corresponding mixture component

$$\mathcal{L}(P_i, \hat{f}_i, f_t) \le \epsilon$$



Key learnwares:
$0.4P_1 + 0.6P_2 = P_t$
$\mathcal{L}(P_1, \hat{f}_1, f_t) \le \epsilon$
$\mathcal{L}(P_2, \hat{f}_1, f_t) \le \epsilon$
Other learnwares:
$w_i = 0$

Solution: **Reweight** & **Compare**



Our method: For learnware $(\tilde{\mu}_i, \hat{f}_i)$ (RKME, Model), corresponds to user's data $\{\boldsymbol{x}_{tn}, y_{tn}\}_{n=1}^{N_t}$

- **Reweight:**
  - Reweight the user's samples to simulate learnware's distribution in RKHS
  - Get a weighted dataset $\{\eta_{in}, (\boldsymbol{x}_{tn}, y_{tn})\}_{n=1}^{N_t}$ with RKME defined as $\tilde{\mu}_{t\to i}$
- **Compare:** on this reweighted dataset,

$$h_i = U\|\tilde{\mu}_i - \tilde{\mu}_{t\to i}\|_{\mathcal{H}} + \sum_{n=1}^{N_t} \eta_{in} L(\hat{f}_i(\boldsymbol{x}_{tn}), y_{tn})$$

- Judge: $h_i \le \theta$: helpful; $h_i > \theta$: unhelpful

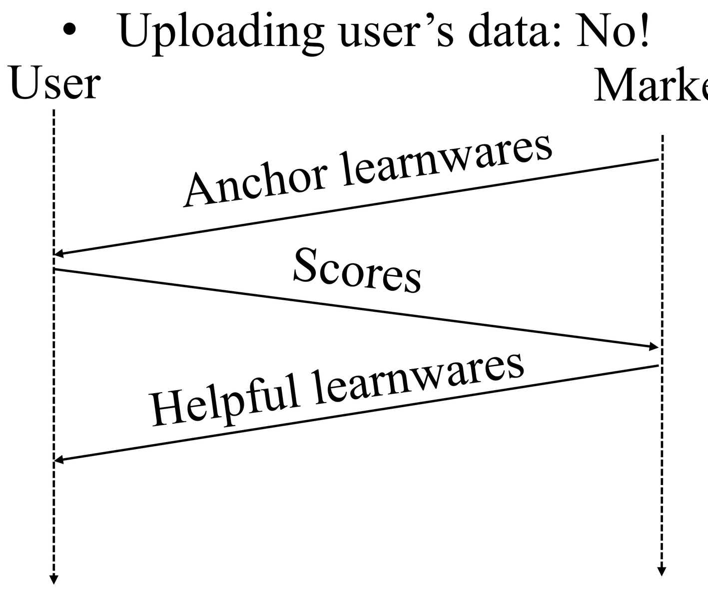
Analyses: There exists a good θ, with a high probability,

- All key learnwares are considered helpful;
- And all learnwares considered helpful can solve one part of user's task.

4. Anchor-based Learnware Identification Framework

Question 2: how to identify helpful learnwares efficiently?

- Examining the whole market: No!
- Uploading user's data: No!



Anchor Learnwares!

1. Market sends anchor learnwares to user;
2. User tests anchor learnwares and returns scores to market;
3. Market identifies helpful learnwares based on scores.

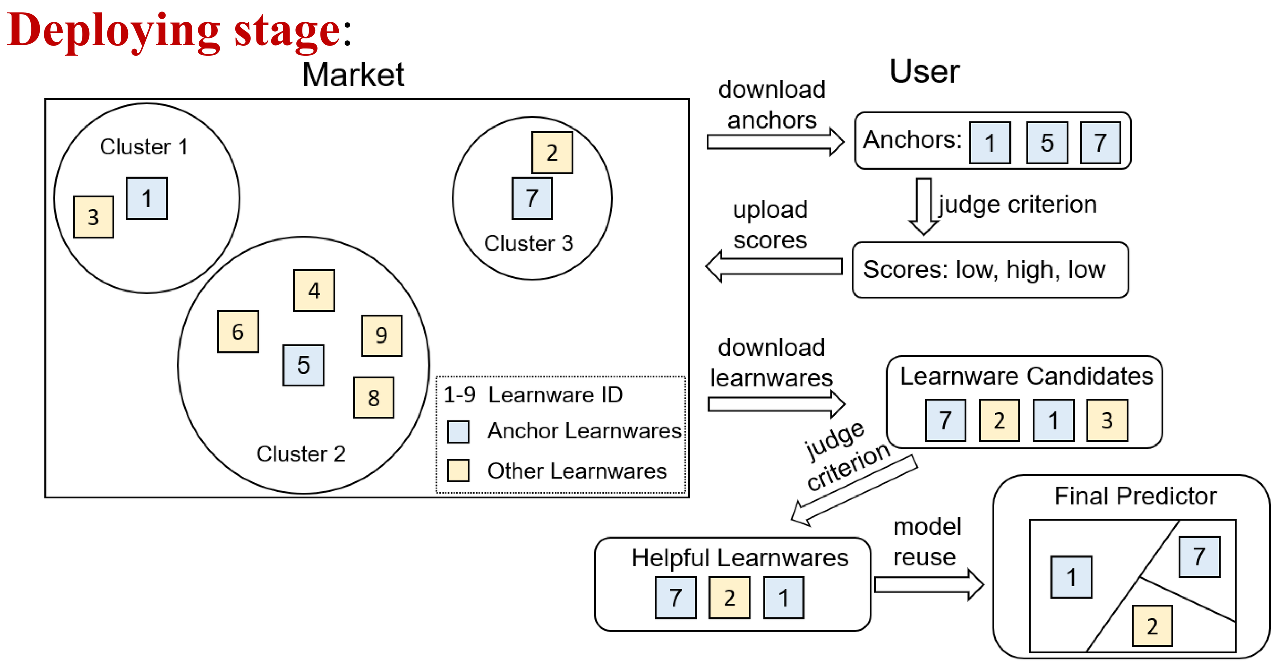
Submitting stage: **Cluster of learnwares**

- A helpful anchor + a good cluster ⟷ The whole cluster may be helpful
- Define a dissimilarity

$$d_{ij} = U \cdot d(P_i, P_j) + min\{\mathcal{L}(P_i, \hat{f}_i, \hat{f}_j), \mathcal{L}(P_j, \hat{f}_i, \hat{f}_j)\}$$

  and then transform to the RKME version.
- Cluster algorithm:
  - PAM: a k-medoids algorithm, medoids as anchors.
  - Multi-level clustering is available for very large markets.
- **Analyses**: helpfulness on user's task: Informally, for any user's task, $|help(learnware) - help(anchor)| \le radius$.

Deploying stage:



5. Experiments

Market construction:

- 4 real-world datasets that can be naturally divided into several parts;
- Train 15 models on each part: different linear models, LightGBM, and neural networks.

Dataset	Task	#Instance	Split Criterion	#Models	#Users
M5	Regression	46M	Department	1050	10
PFS	Regression	9M	Shop	795	17
PPG-DaLiA	Regression	517K	Activity	675	22
Covtype	Classification	581K	Soil	450	10

Results:

1. Our learnware scoring criterion (Ours-traversal) achieve the best performance;
2. Our anchor method (Ours-anchor) greatly improves efficiency (examine 11.8%, 14.9%, 21.42%, 19.91% learnwares) with very little performance degradation.

	M5			PFS			PPG-DaLiA			Covtype		
	RMSE	Imp.	Time	RMSE	Imp.	Time	MSE	Imp.	Time	Error	Imp.	Time
From-scratch	4.142	1.85%	-	3.081	13.79%	-	19.83	45.43%	-	0.334	50.60%	-
Random	4.085	0.00%	-	3.297	0.00%	-	36.62	0.00%	-	0.683	0.00%	-
RKME-task	3.389	18.27%	7.77	2.798	25.42%	2.35	24.53	33.64%	**0.73**	0.380	44.05%	**0.30**
RKME-instance	3.586	13.72%	137.57	2.931	14.56%	277.10	22.40	38.51%	201.42	0.240	65.00%	21.33
Validate	3.266	21.12%	4.09	2.671	29.46%	3.34	14.70	59.87%	10.07	0.245	64.01%	2.43
Ours-traversal	3.154	23.80%	10.61	**2.609**	**32.48%**	8.27	**13.29**	**63.71%**	11.35	**0.222**	**67.67%**	4.94
Ours-anchor	**3.148**	**23.80%**	**1.13**	2.616	32.07%	**1.37**	14.03	61.71%	2.57	0.244	64.45%	1.12

5. Conclusion

1. Propose a novel learnware scoring criterion based on the RKME specification to assess the potential helpfulness of a learnware;
2. Design an anchor-based framework to achieve efficient learnware identification by examining only a small portion of learnwares in the market;
3. Theoretical guarantees + Experimental verification.