# Meta-learning for mixed linear regression

Weihao Kong, Raghav Somani, Zhao Song, Sham Kakade, Sewoong Oh

Presented by: Manaswini Nagaraj

# Motivation and Goals

Motivation:

- Scarcity of large amount of labelled data
- Abundance of number of tasks
- Heterogeneous data from multiple sources

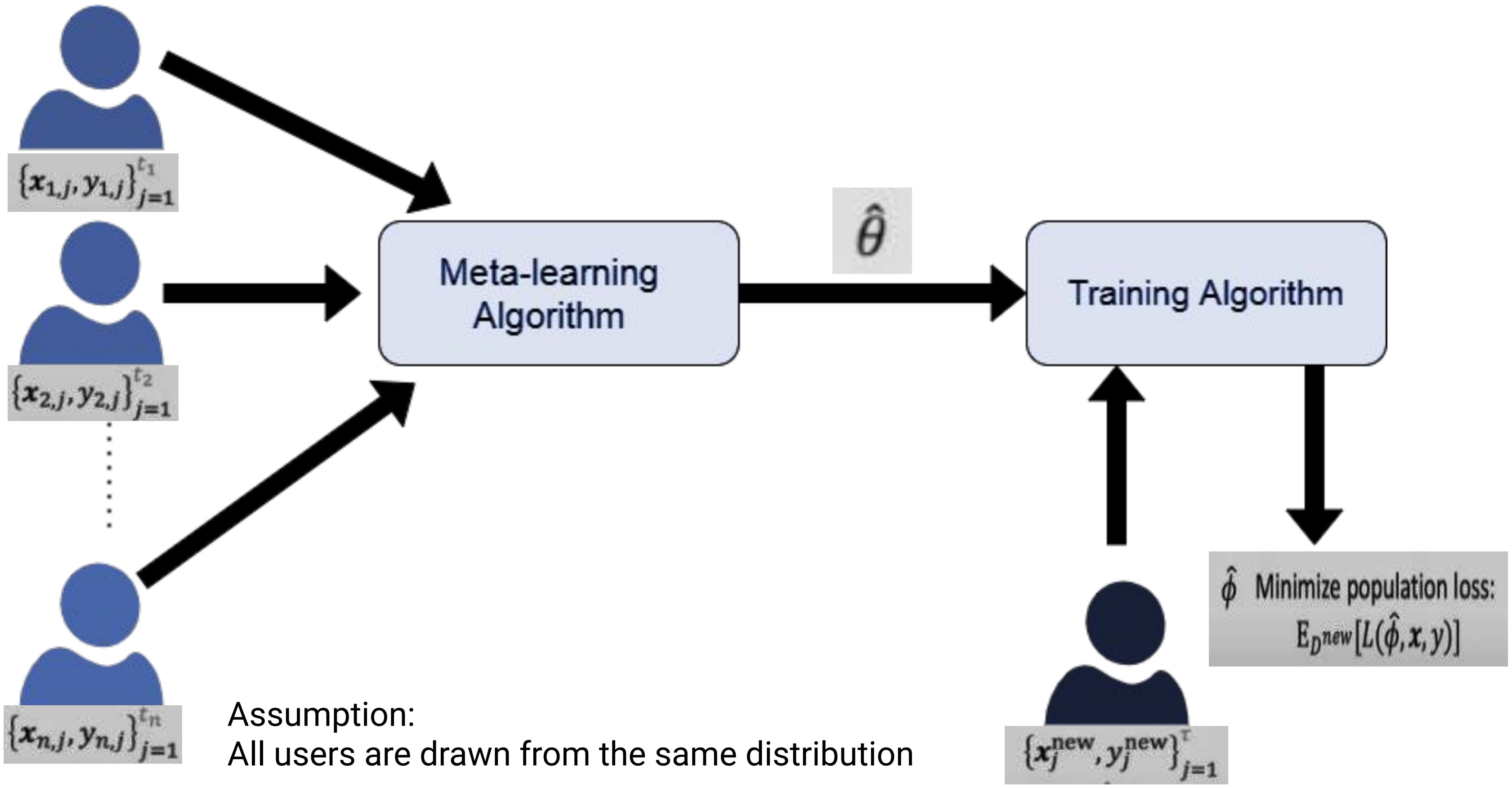
Goals:

- Learn population of the models/parameters
- Use population knowledge to improve new user's model

**Key Question**

**When can abundant tasks with small data compensate for lack of tasks with big data?**

# Meta-learning




Assumption:
All users are drawn from the same distribution

# Probabilistic view on meta-learning

- Collection of n meta-learning tasks {T_i} where i = 1...n are drawn from a distribution P(T)
- Meta-training dataset $\mathcal{D}_{\text{meta-train}} = \{\{(\mathbf{x}_{i,j}, y_{i,j}) \in \mathbb{R}^d \times \mathbb{R}\}_{j \in [t_i]}\}_{i \in [n]}$
- Goal: Train a model for new task T_new coming from the distribution P(T)
- Having only small amount of training data $\mathcal{D} = \left\{(\mathbf{x}_j^{\text{new}}, y_j^{\text{new}})\right\}_{j \in [\tau]}$
- Model parameter Φ_i for each task T_i and meta parameter θ, such that

$$\phi_i \sim \mathbb{P}_\theta(\phi).$$

- The meta-learning problem is defined as estimating the most likely meta-parameter given meta-training data by solving

$$\theta^* \in \arg\max_\theta \log \mathbb{P}(\theta \,|\, \mathcal{D}_{\text{meta-data}})$$

# Prediction after meta-learning

- Once meta-learning is done, the model parameter of a newly arriving task can be estimated by a Maximum a Posteriori (MAP) estimator

$$\widehat{\phi} \in \arg\max_{\phi} \ \log \mathbb{P}(\phi \,|\, \mathcal{D}, \theta^*)$$

- or a Bayes optimal estimator

$$\widehat{\phi} \in \arg\min_{\phi} \ \mathbb{E}_{\phi' \sim \mathbb{P}(\phi' \,|\, \mathcal{D}, \theta^*)}[\,\ell(\phi, \phi')\,]$$

- This estimated parameter is then used for predicting the label of a new data point x in task T_new as

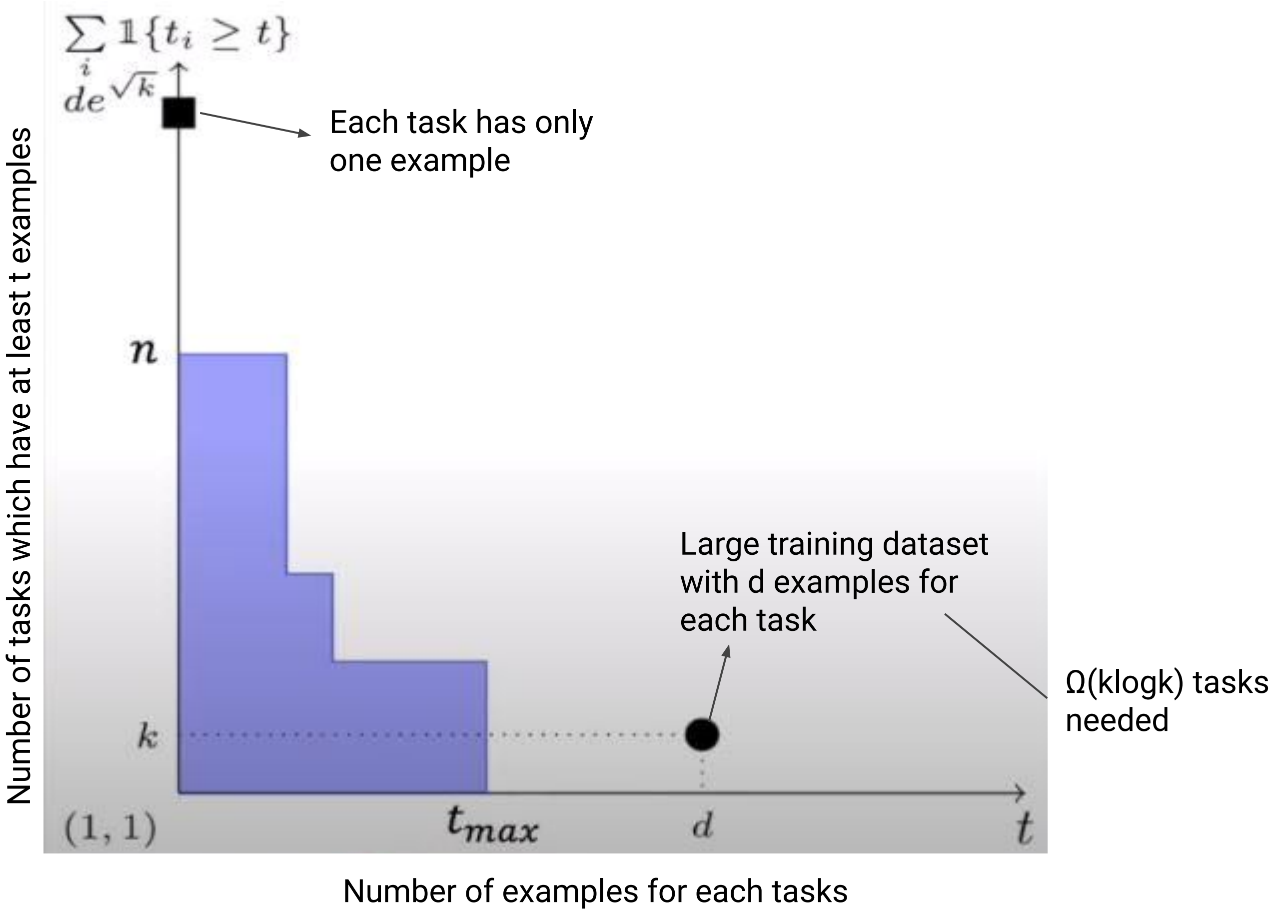
$$\widehat{y} \ \in \ \arg\max_{y} \ \mathbb{P}_{\widehat{\phi}}(y|\mathbf{x})$$

# Mixture of k Linear Regressions

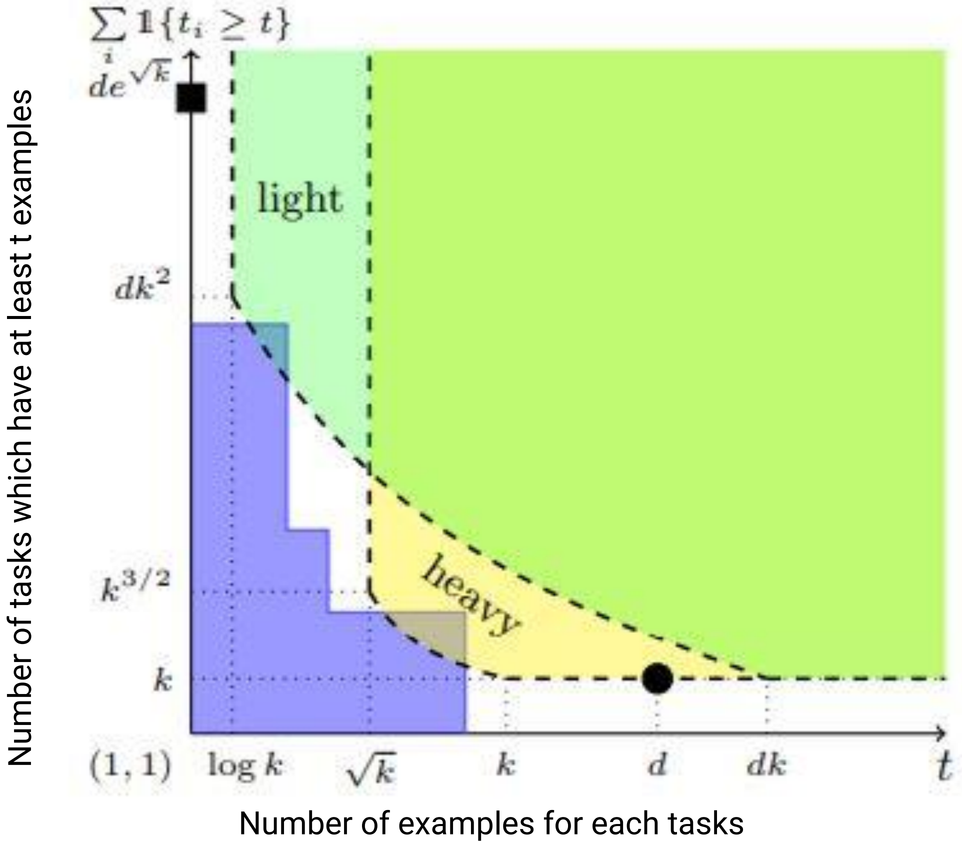
- The meta-learning problem mentioned before is computationally intractable
- So to investigate the trade-offs involved, we assume a simple scenario where tasks are linear regressions $\mathbf{x}_{i,j} \sim \mathcal{P}_{\mathbf{x}}$ , $y_{i,j} = \beta_i^\top \mathbf{x}_{i,j} + \epsilon_{i,j}$
- Here $\beta_i$ is drawn uniformly at randomly from regression vectors {w_1....w_n} $\in \mathbb{R}^d$

- **Main Goals:**
  - Learn distribution of regression vectors
  - Use knowledge of that distribution to improve estimates of new model

# Number of tasks vs Number of examples



# What is required?



# Algorithm Overview

## Subspace Estimation

Compute subspace spanned by the regression vectors using light tasks with singular value decomposition

## Clustering

Project heavy tasks onto the subspace and perform distance based k clustering and estimate w_i

## Classification

Perform likelihood-based classification of light tasks using the estimates from clustering and compute better estimates

# Algorithm

**Algorithm 1**

**Meta-learning**

1. *Subspace estimation.* Compute subspace $\mathbf{U}$ which approximates span $\{\mathbf{w}_1, \ldots, \mathbf{w}_k\}$, with singular value decomposition.

2. *Clustering.* Project the heavy tasks onto the subspace of $\mathbf{U}$, perform distance-based $k$ clustering, and estimate $\widetilde{\mathbf{w}}_i$ for each cluster.

3. *Classification.* Perform likelihood-based classification of the light tasks using $\widetilde{\mathbf{w}}_i$ estimated from the *Clustering* step, and compute the more refined estimates $(\widehat{\mathbf{w}}_i, \widehat{s}_i, \widehat{p}_i)$ of $(\mathbf{w}_i, s_i, p_i)$ for $i \in [k]$.

**Prediction**

4. *Prediction.* Perform MAP or Bayes optimal prediction using the estimated meta-parameter as a prior.

# Subspace Estimation

WLOG, assuming $\mathrm{E}\left[\boldsymbol{x}_{i,j}\boldsymbol{x}_{i,j}^\top\right] = \mathbf{I}_d$. Define random index $z_i$ s.t. $\boldsymbol{w}_{z_i} = \boldsymbol{\beta}_i$.

- Observation: $\mathbf{E}\left[y_{i,1}\boldsymbol{x}_{i,1}|z_i\right] = \mathbf{E}\left[(\boldsymbol{\beta}_i^\top \boldsymbol{x}_{i,1} + \epsilon_{i,1})\boldsymbol{x}_{i,j}\right] = \boldsymbol{\beta}_i$
- Unbiased estimator: $\mathbf{E}\left[y_{i,1}y_{i,2}\boldsymbol{x}_{i,1}\boldsymbol{x}_{i,2}^\top\right] = \mathbf{E}\left[\boldsymbol{\beta}_i\boldsymbol{\beta}_i^\top\right] = \frac{1}{k}\sum_{j=1}^k \boldsymbol{w}_j\boldsymbol{w}_j^\top$
- Subspace: $\mathbf{U} = \mathrm{span}(\boldsymbol{w}_1, \ldots, \boldsymbol{w}_k) = \mathrm{col}\left(\sum_{j=1}^k \boldsymbol{w}_j\boldsymbol{w}_j^\top\right)$

Sample complexity: $dk^2, t \geq 2$

Assumption: k << d

# Clustering

- $(\mathbf{U}x_{i,j}, y_{i,j}) \in \mathbb{R}^k \times \mathbb{R}$ becomes a $k$-dim regression problem.
  Users with $t_i = \Omega(k)$ can learn $\boldsymbol{\beta}_i$ on their own (no need for clustering).
- What if $t_i \ll k$?

**Proposition.** [Kong, Valiant, Brunskill 20] Given two distribution over examples $(\boldsymbol{x}_1, y_1)$ and $(\boldsymbol{x}_2, y_2) \in \mathbb{R}^k \times \mathbb{R}$ such that
- $\mathrm{E}[\boldsymbol{x}_1 \boldsymbol{x}_1^\top] = \mathbf{I}_k,\ y_1 = \beta_1^\top \boldsymbol{x}_1 + noise$
- $\mathrm{E}[\boldsymbol{x}_2 \boldsymbol{x}_2^\top] = \mathbf{I}_k,\ y_2 = \beta_2^\top \boldsymbol{x}_2 + noise$

Can estimate $\beta_1^\top \beta_2$, with $O(\sqrt{k})$ examples from each distribution.
Hint: $\mathrm{E}[y_1 y_2 \boldsymbol{x}_1^\top \boldsymbol{x}_2] = \beta_1^\top \beta_2$

- Can determine whether two users share the same regression vector with $t_i = O(\sqrt{k})$
- $k$-cluster the users and roughly estimate $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_k$
- Sample complexity: $O(k^2), t \geq \sqrt{k}$

# Classification

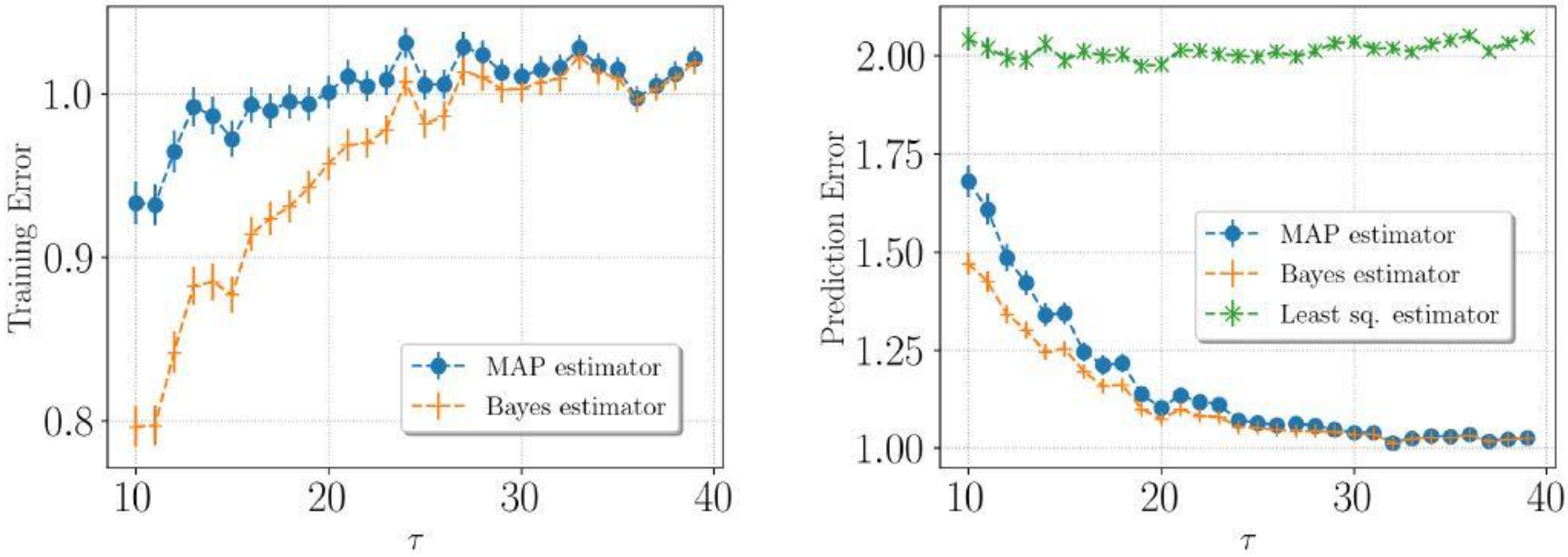
- Intuition: once having rough estimate of $\boldsymbol{w}_1, \dots \boldsymbol{w}_k$, easy to determine which is $\beta_i$ (only $t = \log k$ needed, instead of $\sqrt{k}$).

- For all $l = 1 \dots k$,

$$\mathrm{Var}\left[y_{i,1} - \boldsymbol{x}_{i,1}^{\top}\boldsymbol{w}_l\right] = \mathrm{Var}[noise] + \|\boldsymbol{\beta}_i - \boldsymbol{w}_l\|_2^2$$

Var is large when $l \neq z_i$!! ($\boldsymbol{\beta}_i = \boldsymbol{w}_{z_i}$)

- Need $t = \log k$ to make sure $w_{z_i}$ has the smallest residual (union bound).
- Sample complexity: $t \geq \log k$ , $dk/\epsilon^2$ total.




(a) Training error

(b) Prediction error

# Conclusion and Future scope of work

**Conclusion:**

The proposed algorithm will efficiently utilize light task data as long as there exists some heavy task data too, each with at least $\sqrt{k}$ examples.

**Future scope of work:**

- Investigate the case where $t_H = o(\sqrt{k})$
- What happens if $P_x$ is different for different tasks?
- Application of the method beyond regression.