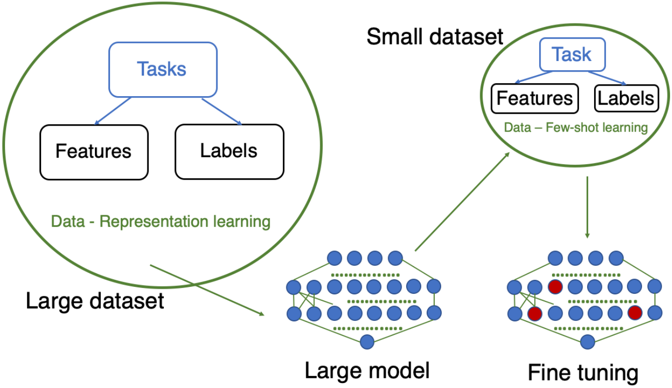
Subspace Based Meta-Learning

Yue Sun
University of Washington

Joint work with:
Halil Ibrahim Gulluk (Bogazici University), Adhyyan Narang (Univ. of Washington), Samet Oymak (UC Riverside), Maryam Fazel (Univ. of Washington)

April 9, 2021

Meta learning



Meta learning

Task, feature in $\mathbb{R}^d$, label in $\mathbb{R}$.
Task: $\boldsymbol{\beta} \sim \mathcal{N}(0, \boldsymbol{\Sigma}_{\boldsymbol{\beta}})$, $\boldsymbol{\Sigma}_{\boldsymbol{\beta}}$ approx low rank, **Feature:** $\boldsymbol{x} \sim \mathcal{N}(0, \boldsymbol{\Sigma}_{\boldsymbol{x}})$,
Noise: $\varepsilon \sim \mathcal{N}(0, \sigma_\varepsilon)$, **Label:** $y = \boldsymbol{x}^\top \boldsymbol{\beta} + \varepsilon$.

- **Two steps:** Representation learning, Few-shot learning

Meta learning

Task, feature in $\mathbb{R}^d$, label in $\mathbb{R}$.
Task: $\boldsymbol{\beta} \sim \mathcal{N}(0, \boldsymbol{\Sigma}_{\boldsymbol{\beta}})$, $\boldsymbol{\Sigma}_{\boldsymbol{\beta}}$ approx low rank, **Feature:** $\boldsymbol{x} \sim \mathcal{N}(0, \boldsymbol{\Sigma}_{\boldsymbol{X}})$,
Noise: $\varepsilon \sim \mathcal{N}(0, \sigma_\varepsilon)$, **Label:** $y = \boldsymbol{x}^\top \boldsymbol{\beta} + \varepsilon$.

- **Two steps:** Representation learning, Few-shot learning
- **Rep learning:** Sample $\boldsymbol{\beta}_1, ..., \boldsymbol{\beta}_{n_{\text{task}}}$. For $i \in [n_{\text{task}}]$, Sample $\boldsymbol{x}_{i,1}, ..., \boldsymbol{x}_{i,n_{\text{spt}}}$. Evaluate y. Use x, y to estimate $\boldsymbol{\Sigma}_{\boldsymbol{\beta}}$.

Meta learning

Task, feature in $\mathbb{R}^d$, label in $\mathbb{R}$.
Task: $\boldsymbol{\beta} \sim \mathcal{N}(0, \boldsymbol{\Sigma}_{\boldsymbol{\beta}})$, $\boldsymbol{\Sigma}_{\boldsymbol{\beta}}$ approx low rank, **Feature:** $\boldsymbol{x} \sim \mathcal{N}(0, \boldsymbol{\Sigma}_{\boldsymbol{X}})$,
Noise: $\varepsilon \sim \mathcal{N}(0, \sigma_\varepsilon)$, **Label:** $y = \boldsymbol{x}^\top \boldsymbol{\beta} + \varepsilon$.

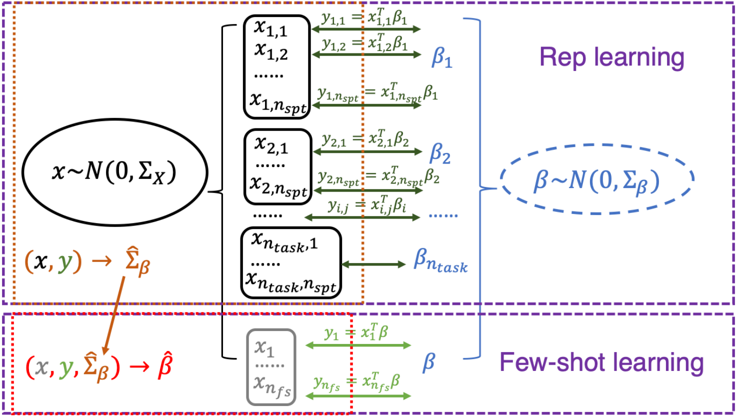
- **Two steps:** Representation learning, Few-shot learning
- **Rep learning:** Sample $\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_{n_{\mathrm{task}}}$. For $i \in [n_{\mathrm{task}}]$, Sample $\boldsymbol{x}_{i,1}, \ldots, \boldsymbol{x}_{i,n_{\mathrm{spt}}}$. Evaluate y. Use x, y to estimate $\boldsymbol{\Sigma}_{\boldsymbol{\beta}}$.
- **Few-shot learning:** Sample $\boldsymbol{\beta}$, $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{n_{\mathrm{spt}}}$, evaluate y. Use x, y and estimate $\boldsymbol{\beta}$ in principal subspace of $\boldsymbol{\Sigma}_{\boldsymbol{\beta}}$.

Meta learning

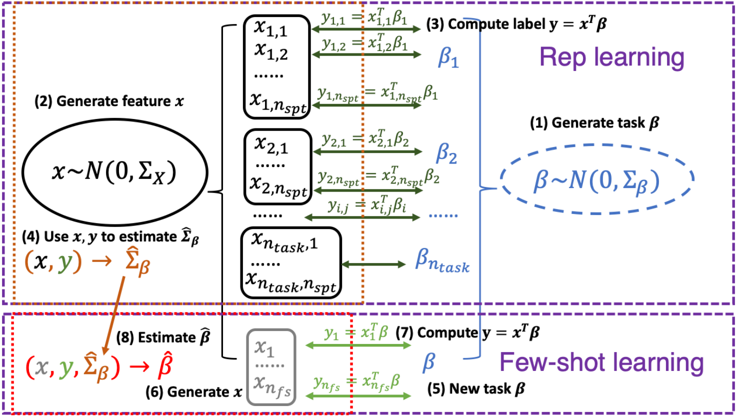
Task, feature in $\mathbb{R}^d$, label in $\mathbb{R}$.
Task: $\boldsymbol{\beta} \sim \mathcal{N}(0, \boldsymbol{\Sigma}_{\boldsymbol{\beta}})$, $\boldsymbol{\Sigma}_{\boldsymbol{\beta}}$ approx low rank, **Feature:** $\boldsymbol{x} \sim \mathcal{N}(0, \boldsymbol{\Sigma}_{\boldsymbol{X}})$,
Noise: $\varepsilon \sim \mathcal{N}(0, \sigma_\varepsilon)$, **Label:** $y = \boldsymbol{x}^\top \boldsymbol{\beta} + \varepsilon$.

- **Two steps:** Representation learning, Few-shot learning
- **Rep learning:** Sample $\boldsymbol{\beta}_1, ..., \boldsymbol{\beta}_{n_{\mathrm{task}}}$. For $i \in [n_{\mathrm{task}}]$, Sample $\boldsymbol{x}_{i,1}, ..., \boldsymbol{x}_{i,n_{\mathrm{spt}}}$. Evaluate y. Use x, y to estimate $\boldsymbol{\Sigma}_{\boldsymbol{\beta}}$.
- **Few-shot learning:** Sample $\boldsymbol{\beta}$, $\boldsymbol{x}_1, ..., \boldsymbol{x}_{n_{\mathrm{spt}}}$, evaluate y. Use x, y and estimate $\boldsymbol{\beta}$ in principal subspace of $\boldsymbol{\Sigma}_{\boldsymbol{\beta}}$.
- **Few-shot learning:** Sample $\boldsymbol{\beta}$, $\boldsymbol{x}_1, ..., \boldsymbol{x}_{n_{\mathrm{fs}}}$, evaluate y. Use x, y and a shaping matrix as a function of $\hat{\boldsymbol{\Sigma}}_{\boldsymbol{\beta}}$ to estimate $\boldsymbol{\beta}$.

Meta learning



Meta learning



Meta-learning - Linear - Prior works

- Mei & Montanari. Double descent.
- Du et al. Matrix factorization type. No algorithm.
- Kong et al. Method of moment (MoM) estimator. $O(dr^2)$ samples for rep learning, $O(r)$ samples for few-shot learning.
- Tripuraneni et al. MoM estimator and gradient descent. MoM: same sample complexity as above. GD: $O(dr^4)$ samples for rep learning
- Bartlett et al., Wu & Xu, Nakkiran et al. Overparameterized few-shot learning via optimal ridge regularization.

Overview

Representation learning - Linear

Few-shot learning - Linear

Meta learning - Nonlinear

Rep learning - learn $\boldsymbol{\Sigma}_{\boldsymbol{\beta}}$

Task: $\boldsymbol{\beta} \sim \mathcal{N}(0, \boldsymbol{\Sigma}_{\boldsymbol{\beta}})$, **Feature:** $\boldsymbol{x} \sim \mathcal{N}(0, \boldsymbol{\Sigma}_{\boldsymbol{X}})$, **Label:** $y = \boldsymbol{x}^\top \boldsymbol{\beta}$. Suppose $\mathrm{rank}(\boldsymbol{\Sigma}_{\boldsymbol{X}}) = r_f$, $\mathrm{rank}(\boldsymbol{\Sigma}_{\boldsymbol{\beta}}) = r_t$. Suppose the principal subspaces of $\boldsymbol{\Sigma}_{\boldsymbol{X}}$ and $\boldsymbol{\Sigma}_{\boldsymbol{\beta}}$ align.

Rep learning: Sample $\boldsymbol{\beta}_1, ..., \boldsymbol{\beta}_{n_{\mathrm{task}}}$. For $i \in [n_{\mathrm{task}}]$, Sample $\boldsymbol{x}_{i,1}, ..., \boldsymbol{x}_{i,n_{\mathrm{spt}}}$. Evaluate y. Use x, y to estimate $\boldsymbol{\Sigma}_{\boldsymbol{\beta}}$.

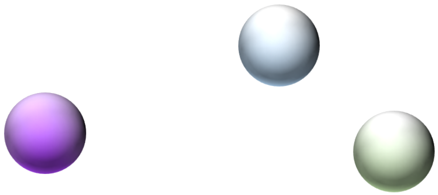
Rep learning - learn $\boldsymbol{\Sigma}_{\boldsymbol{\beta}}$

Task: $\boldsymbol{\beta} \sim \mathcal{N}(0, \boldsymbol{\Sigma}_{\boldsymbol{\beta}})$, **Feature:** $\boldsymbol{x} \sim \mathcal{N}(0, \boldsymbol{\Sigma}_{\boldsymbol{X}})$, **Label:** $y = \boldsymbol{x}^\top \boldsymbol{\beta}$.
Suppose $\mathrm{rank}(\boldsymbol{\Sigma}_{\boldsymbol{X}}) = r_f$, $\mathrm{rank}(\boldsymbol{\Sigma}_{\boldsymbol{\beta}}) = r_t$. Suppose the principal subspaces of $\boldsymbol{\Sigma}_{\boldsymbol{X}}$ and $\boldsymbol{\Sigma}_{\boldsymbol{\beta}}$ align.

Rep learning: Sample $\boldsymbol{\beta}_1, ..., \boldsymbol{\beta}_{n_{\mathrm{task}}}$. For $i \in [n_{\mathrm{task}}]$, Sample $\boldsymbol{x}_{i,1}, ..., \boldsymbol{x}_{i,n_{\mathrm{spt}}}$. Evaluate y. Use x, y to estimate $\boldsymbol{\Sigma}_{\boldsymbol{\beta}}$.

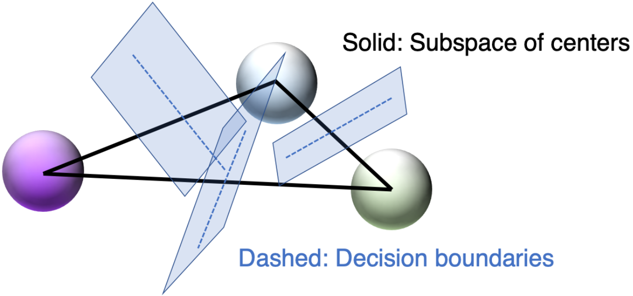
Interested in **feature-task alignment**.
E.g., $\boldsymbol{\Sigma}_{\boldsymbol{\beta}} = \mathrm{diag}(\boldsymbol{I}_{r_t}, 0)$, $\boldsymbol{\Sigma}_{\boldsymbol{X}} = \mathrm{diag}(\boldsymbol{I}_{r_f}, \iota \boldsymbol{I}_{d-r_f})$.

Motivating example: Multi-class classification

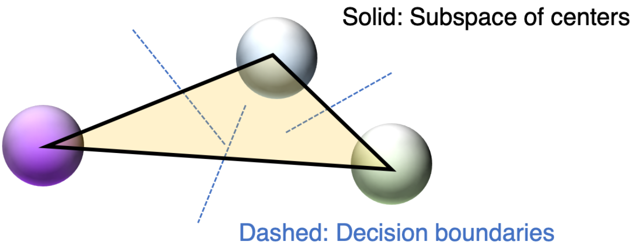


Motivation: classification of Gaussian mixture

Motivating example: Multi-class classification



Motivating example: Multi-class classification



$$\dim\big(\mathrm{span}(\mathrm{centers})\big) < \dim(\mathrm{space})$$

Naive case: estimating $\boldsymbol{\Sigma}_{\boldsymbol{X}}$ is enough

1. $\boldsymbol{\Sigma}_{\boldsymbol{\beta}} = \boldsymbol{\Sigma}_{\boldsymbol{X}}$.
2. $\mathrm{span}(\boldsymbol{\Sigma}_{\boldsymbol{\beta}}) = \mathrm{span}(\boldsymbol{\Sigma}_{\boldsymbol{X}})$.
3. $\mathrm{span}(\boldsymbol{\Sigma}_{\boldsymbol{\beta}}) \subset \mathrm{span}(\boldsymbol{\Sigma}_{\boldsymbol{X}})$ but we are satisfied with $\mathrm{span}(\boldsymbol{\Sigma}_{\boldsymbol{X}})$.

Naive case: estimating $\boldsymbol{\Sigma}_{\boldsymbol{X}}$ is enough

1. $\boldsymbol{\Sigma}_{\boldsymbol{\beta}} = \boldsymbol{\Sigma}_{\boldsymbol{X}}$.
2. $\mathrm{span}(\boldsymbol{\Sigma}_{\boldsymbol{\beta}}) = \mathrm{span}(\boldsymbol{\Sigma}_{\boldsymbol{X}})$.
3. $\mathrm{span}(\boldsymbol{\Sigma}_{\boldsymbol{\beta}}) \subset \mathrm{span}(\boldsymbol{\Sigma}_{\boldsymbol{X}})$ but we are satisfied with $\mathrm{span}(\boldsymbol{\Sigma}_{\boldsymbol{X}})$.

MoM-F estimator:

$$\hat{\boldsymbol{\Sigma}}_{\boldsymbol{X}} = \frac{1}{n_{\mathrm{tot}}} \sum_{j=1}^{n_{\mathrm{spt}}} \sum_{i=1}^{n_{\mathrm{task}}} \boldsymbol{x}_{ij} \boldsymbol{x}_{ij}^{\top}$$

Sample complexity: $\mathcal{O}(r_f)$. Error: $\mathcal{O}(\sqrt{r_f / n_{\mathrm{tot}}})$.

General case: MoM estimator

When n_{spt} is small.

$$\hat{\boldsymbol{Q}} = \frac{1}{n_{\mathrm{task}}} \sum_{i=1}^{n_{\mathrm{task}}} \frac{1}{n_{\mathrm{spt}}} \sum_{j=1}^{n_{\mathrm{spt}}} y_{ij}^2 \boldsymbol{x}_{ij} \boldsymbol{x}_{ij}^\top .$$

$$\hat{\boldsymbol{M}} = \frac{1}{n_{\mathrm{task}}} \sum_{i=1}^{n_{\mathrm{task}}} \frac{2}{n_{\mathrm{spt}}^2} \left[\sum_{j=1}^{n_{\mathrm{spt}}/2} y_{ij} y_{i(j+n_{\mathrm{spt}}/2)} \cdot (\boldsymbol{x}_{ij} \boldsymbol{x}_{i(j+n_{\mathrm{spt}}/2)}^\top + \boldsymbol{x}_{i(j+n_{\mathrm{spt}}/2)} \boldsymbol{x}_{ij}^\top) \right] .$$

General case: MoM estimator

When n_{spt} is small.

$$\hat{\boldsymbol{Q}} = \frac{1}{n_{\mathrm{task}}} \sum_{i=1}^{n_{\mathrm{task}}} \frac{1}{n_{\mathrm{spt}}} \sum_{j=1}^{n_{\mathrm{spt}}} y_{ij}^2 \boldsymbol{x}_{ij} \boldsymbol{x}_{ij}^\top .$$

$$\hat{\boldsymbol{M}} = \frac{1}{n_{\mathrm{task}}} \sum_{i=1}^{n_{\mathrm{task}}} \frac{2}{n_{\mathrm{spt}}^2} \left[\sum_{j=1}^{n_{\mathrm{spt}}/2} y_{ij} y_{i(j+n_{\mathrm{spt}}/2)} \cdot (\boldsymbol{x}_{ij} \boldsymbol{x}_{i(j+n_{\mathrm{spt}}/2)}^\top + \boldsymbol{x}_{i(j+n_{\mathrm{spt}}/2)} \boldsymbol{x}_{ij}^\top) \right] .$$

Mean:
$\boldsymbol{Q} = 2\boldsymbol{\Sigma}_{\boldsymbol{X}} \boldsymbol{\Sigma}_{\boldsymbol{\beta}} \boldsymbol{\Sigma}_{\boldsymbol{X}} + \mathrm{tr}(\boldsymbol{\Sigma}_{\boldsymbol{\beta}} \boldsymbol{\Sigma}_{\boldsymbol{X}}) \boldsymbol{\Sigma}_{\boldsymbol{X}}$.
$\boldsymbol{M} = \boldsymbol{\Sigma}_{\boldsymbol{X}} \boldsymbol{\Sigma}_{\boldsymbol{\beta}} \boldsymbol{\Sigma}_{\boldsymbol{X}}$.

Sample complexity: $\mathcal{O}(r_f r_t^2)$. Error: $\mathcal{O}(\sqrt{r_f r_t^2 / n_{\mathrm{tot}}} + \sqrt{r_t / n_{\mathrm{task}}})$.

General case: MoM-TA estimator

We first define $\hat{\boldsymbol{b}}_i = \sum_{j=1}^{n_{\rm spt}} y_{ij}\boldsymbol{x}_{ij}$, for every $i = 1, ..., n_{\rm task}$.

$$\hat{\boldsymbol{B}} = [\hat{\boldsymbol{b}}_1, ..., \hat{\boldsymbol{b}}_{n_{\rm task}}],$$
$$\hat{\boldsymbol{G}} = n_{\rm task}^{-1}\hat{\boldsymbol{B}}\hat{\boldsymbol{B}}^\top.$$

General case: MoM-TA estimator

We first define $\hat{\boldsymbol{b}}_i = \sum_{j=1}^{n_{\mathrm{spt}}} y_{ij}\boldsymbol{x}_{ij}$, for every $i = 1, ..., n_{\mathrm{task}}$.

$$\begin{aligned}\hat{\boldsymbol{B}} &= [\hat{\boldsymbol{b}}_1, ..., \hat{\boldsymbol{b}}_{n_{\mathrm{task}}}],\\ \hat{\boldsymbol{G}} &= n_{\mathrm{task}}^{-1}\hat{\boldsymbol{B}}\hat{\boldsymbol{B}}^\top.\end{aligned}$$

Mean:

$$\boldsymbol{G} = \boldsymbol{\Sigma}_{\boldsymbol{X}}\boldsymbol{\Sigma}_{\boldsymbol{\beta}}\boldsymbol{\Sigma}_{\boldsymbol{X}} + n_{\mathrm{spt}}^{-1}(\boldsymbol{\Sigma}_{\boldsymbol{X}}\boldsymbol{\Sigma}_{\boldsymbol{\beta}}\boldsymbol{\Sigma}_{\boldsymbol{X}} + \mathrm{tr}(\boldsymbol{\Sigma}_{\boldsymbol{\beta}}\boldsymbol{\Sigma}_{\boldsymbol{X}})\boldsymbol{\Sigma}_{\boldsymbol{X}})$$

Sample complexity:
1. Generally $\mathcal{O}(r_f r_t^2)$.
2. $\mathcal{O}(r_f r_t)$ when $n_{\mathrm{spt}} \geq r_t$.

Rep learning - learn $\boldsymbol{\Sigma}_{\boldsymbol{\beta}}$

Task: $\boldsymbol{\beta} \sim \mathcal{N}(0, \boldsymbol{\Sigma}_{\boldsymbol{\beta}})$, **Feature:** $\boldsymbol{x} \sim \mathcal{N}(0, \boldsymbol{\Sigma}_{\boldsymbol{X}})$, **Label:** $y = \boldsymbol{x}^\top \boldsymbol{\beta}$.
Suppose $\mathrm{rank}(\boldsymbol{\Sigma}_{\boldsymbol{X}}) = r_f$, $\mathrm{rank}(\boldsymbol{\Sigma}_{\boldsymbol{\beta}}) = r_t$. Suppose the principal subspaces of $\boldsymbol{\Sigma}_{\boldsymbol{X}}$ and $\boldsymbol{\Sigma}_{\boldsymbol{\beta}}$ align.
Rep learning: Sample $\boldsymbol{\beta}_1, ..., \boldsymbol{\beta}_{n_{\mathrm{task}}}$. For $i \in [n_{\mathrm{task}}]$, Sample $\boldsymbol{x}_{i,1}, ..., \boldsymbol{x}_{i,n_{\mathrm{spt}}}$. Evaluate y. Use x, y to estimate $\boldsymbol{\Sigma}_{\boldsymbol{\beta}}$.

Estimators: MoM, MoM-TA, MoM-F.
MoM: $\sum_{i,j} y_{ij}^2 \boldsymbol{x}_{ij} \boldsymbol{x}_{ij}^\top$.
MoM-TA: Let $\hat{\boldsymbol{b}}_i = \sum_{j=1}^{n_{\mathrm{spt}}} y_{ij} \boldsymbol{x}_{ij}$. $\hat{\boldsymbol{B}} = [\hat{\boldsymbol{b}}_1, ..., \hat{\boldsymbol{b}}_{n_{\mathrm{task}}}]$. Need $n_{\mathrm{spt}} \geq r_t$.
MoM-F: $\sum_{i,j} \boldsymbol{x}_{ij} \boldsymbol{x}_{ij}^\top$.

$\boldsymbol{\Sigma}_{\boldsymbol{\beta}} = \mathrm{diag}(\boldsymbol{I}_{r_t}, 0)$. Extra $(r_t / n_{\mathrm{task}})^{1/2}$ term in MoM and MoM-TA ignored.

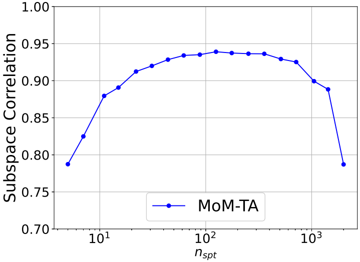
feature cov	$\boldsymbol{\Sigma}_{\boldsymbol{X}} = \boldsymbol{I}$		$\boldsymbol{\Sigma}_{\boldsymbol{X}} = \mathrm{diag}(\boldsymbol{I}_{r_f}, 0)$	
estimator	min sample	error	min sample	error
MoM	dr_t^2	$(dr_t^2 / n_{\mathrm{tot}})^{1/2}$	$r_f r_t^2$	$(r_f r_t^2 / n_{\mathrm{tot}})^{1/2}$
MoM-TA	dr_t	$(r_t / n_{\mathrm{spt}})^{1/2}$	$r_f r_t$	$(r_t / n_{\mathrm{spt}})^{1/2}$
MoM-F	-	-	r_f	$(r_f / n_{\mathrm{tot}})^{1/2}$

Tradeoff between n_{spt} and n_{task}

n_{spt}: Sample per task
n_{task}: Number of tasks
n_{tot}: Total samples $= n_{\mathrm{spt}} n_{\mathrm{task}}$.
Question: Fix n_{tot} and change $n_{\mathrm{spt}}, n_{\mathrm{task}}$.

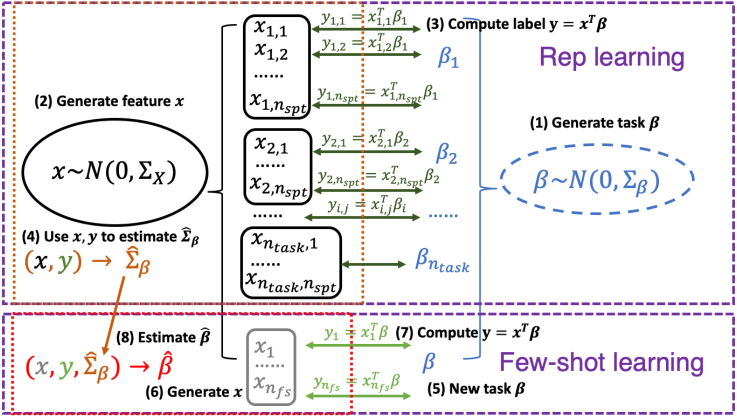
Tradeoff between n_{spt} and n_{task}

n_{spt}: Sample per task
n_{task}: Number of tasks
n_{tot}: Total samples $= n_{\mathrm{spt}} n_{\mathrm{task}}$.
Question: Fix n_{tot} and change $n_{\mathrm{spt}}, n_{\mathrm{task}}$.



Overview

Meta learning



Few-shot learning - learn $\boldsymbol{\beta}$

Task: $\boldsymbol{\beta} \sim \mathcal{N}(0, \boldsymbol{\Sigma}_{\boldsymbol{\beta}})$, **Feature:** $\boldsymbol{x} \sim \mathcal{N}(0, \boldsymbol{\Sigma}_{\boldsymbol{X}})$, **Label:** $y = \boldsymbol{x}^{\top}\boldsymbol{\beta}$.
Suppose $\text{rank}(\boldsymbol{\Sigma}_{\boldsymbol{X}}) = r_f$, $\text{rank}(\boldsymbol{\Sigma}_{\boldsymbol{\beta}}) = r_t$.
Few-shot learning: Sample $\boldsymbol{\beta}$, $\boldsymbol{x}_1, ..., \boldsymbol{x}_{n_{\text{fs}}}$, evaluate y. Use x, y and a shaping matrix as a function of $\hat{\boldsymbol{\Sigma}}_{\boldsymbol{\beta}}$ to estimate $\boldsymbol{\beta}$.

Few-shot learning - learn $\boldsymbol{\beta}$

Task: $\boldsymbol{\beta} \sim \mathcal{N}(0, \boldsymbol{\Sigma}_{\boldsymbol{\beta}})$, **Feature:** $\boldsymbol{x} \sim \mathcal{N}(0, \boldsymbol{\Sigma}_{\boldsymbol{X}})$, **Label:** $y = \boldsymbol{x}^\top \boldsymbol{\beta}$.
Suppose $\operatorname{rank}(\boldsymbol{\Sigma}_{\boldsymbol{X}}) = r_f$, $\operatorname{rank}(\boldsymbol{\Sigma}_{\boldsymbol{\beta}}) = r_t$.
Few-shot learning: Sample $\boldsymbol{\beta}$, $\boldsymbol{x}_1, ..., \boldsymbol{x}_{n_{\text{fs}}}$, evaluate y. Use x, y and a shaping matrix as a function of $\hat{\boldsymbol{\Sigma}}_{\boldsymbol{\beta}}$ to estimate $\boldsymbol{\beta}$.

Prior work: Restrict $\hat{\boldsymbol{\beta}}$ in principal subspace of $\hat{\boldsymbol{\Sigma}}_{\boldsymbol{\beta}}$. Dimension $< n_{\text{fs}}$.

Few-shot learning - learn $\boldsymbol{\beta}$

Task: $\boldsymbol{\beta} \sim \mathcal{N}(0, \boldsymbol{\Sigma}_{\boldsymbol{\beta}})$, **Feature:** $\boldsymbol{x} \sim \mathcal{N}(0, \boldsymbol{\Sigma}_{\boldsymbol{X}})$, **Label:** $y = \boldsymbol{x}^\top \boldsymbol{\beta}$.
Suppose $\mathrm{rank}(\boldsymbol{\Sigma}_{\boldsymbol{X}}) = r_f$, $\mathrm{rank}(\boldsymbol{\Sigma}_{\boldsymbol{\beta}}) = r_t$.
Few-shot learning: Sample $\boldsymbol{\beta}$, $\boldsymbol{x}_1, ..., \boldsymbol{x}_{n_{\mathrm{fs}}}$, evaluate y. Use x, y and a shaping matrix as a function of $\hat{\boldsymbol{\Sigma}}_{\boldsymbol{\beta}}$ to estimate $\boldsymbol{\beta}$.

Prior work: Restrict $\hat{\boldsymbol{\beta}}$ in principal subspace of $\hat{\boldsymbol{\Sigma}}_{\boldsymbol{\beta}}$. Dimension $< n_{\mathrm{fs}}$.

Our work: An arbitrary dimension R, and set a shaping matrix $\Lambda \in \mathbb{R}^{R \times d}$ as a function of $\hat{\boldsymbol{\Sigma}}_{\boldsymbol{\beta}}$ that helps with few-shot learning.

How does shaping matrix work

Task: $\boldsymbol{\beta} \sim \mathcal{N}(0, \boldsymbol{\Sigma}_{\boldsymbol{\beta}})$, **Feature:** $\boldsymbol{x} \sim \mathcal{N}(0, \boldsymbol{\Sigma}_{\boldsymbol{X}})$, **Label:** $y = \boldsymbol{x}^\top \boldsymbol{\beta}$.
Suppose $\text{rank}(\boldsymbol{\Sigma}_{\boldsymbol{X}}) = r_f$, $\text{rank}(\boldsymbol{\Sigma}_{\boldsymbol{\beta}}) = r_t$.
Few-shot learning: Sample $\boldsymbol{\beta}$, $\boldsymbol{x}_1, ..., \boldsymbol{x}_{n_{\text{fs}}}$, evaluate y. Use x, y and a shaping matrix as a function of $\hat{\boldsymbol{\Sigma}}_{\boldsymbol{\beta}}$ to estimate $\boldsymbol{\beta}$.

Min norm solution with Λ.

$$\hat{\boldsymbol{\alpha}}_\Lambda = \arg\min_{\boldsymbol{\alpha}} \|\boldsymbol{\alpha}\|_{\ell_2} \text{ s.t. } \boldsymbol{y} = \boldsymbol{X}\Lambda\boldsymbol{\alpha}$$

$$\hat{\boldsymbol{\beta}}_\Lambda = \Lambda\hat{\boldsymbol{\alpha}}_\Lambda = \Lambda(\boldsymbol{X}\Lambda)^\dagger \boldsymbol{y}.$$

How does shaping matrix work

Task: $\boldsymbol{\beta} \sim \mathcal{N}(0, \boldsymbol{\Sigma}_{\boldsymbol{\beta}})$, **Feature:** $\boldsymbol{x} \sim \mathcal{N}(0, \boldsymbol{\Sigma}_{\boldsymbol{X}})$, **Label:** $y = \boldsymbol{x}^\top \boldsymbol{\beta}$.
Suppose $\text{rank}(\boldsymbol{\Sigma}_{\boldsymbol{X}}) = r_f$, $\text{rank}(\boldsymbol{\Sigma}_{\boldsymbol{\beta}}) = r_t$.
Few-shot learning: Sample $\boldsymbol{\beta}$, $\boldsymbol{x}_1, ..., \boldsymbol{x}_{n_{\mathrm{fs}}}$, evaluate y. Use x, y and a shaping matrix as a function of $\hat{\boldsymbol{\Sigma}}_{\boldsymbol{\beta}}$ to estimate $\boldsymbol{\beta}$.

Min norm solution with Λ.

$$\hat{\boldsymbol{\alpha}}_\Lambda = \arg\min_{\boldsymbol{\alpha}} \|\boldsymbol{\alpha}\|_{\ell_2} \text{ s.t. } \boldsymbol{y} = \boldsymbol{X}\Lambda\boldsymbol{\alpha}$$
$$\hat{\boldsymbol{\beta}}_\Lambda = \Lambda\hat{\boldsymbol{\alpha}}_\Lambda = \Lambda(\boldsymbol{X}\Lambda)^\dagger \boldsymbol{y}.$$

$$\hat{\boldsymbol{\beta}}_\Lambda = \lim_{t \to 0} \mathrm{argmin}_{\boldsymbol{\beta}} \|\boldsymbol{X}^\top \boldsymbol{\beta} - \boldsymbol{y}\|^2 + t\boldsymbol{\beta}^\top \Lambda^{-2} \boldsymbol{\beta}$$

Error metric for asymptotic case

Risk function

$$\begin{aligned}\text{risk}(\Lambda, \boldsymbol{\Sigma}_{\boldsymbol{\beta}}) &= \boldsymbol{E}(y - \boldsymbol{x}^\top \hat{\boldsymbol{\beta}}_\Lambda)^2 \\ &= \boldsymbol{E}(\hat{\boldsymbol{\beta}}_\Lambda - \boldsymbol{\beta})^\top \boldsymbol{\Sigma}_{\boldsymbol{X}} (\hat{\boldsymbol{\beta}}_\Lambda - \boldsymbol{\beta}).\end{aligned}$$

Error metric for asymptotic case

Risk function

$$\begin{aligned}\mathrm{risk}(\Lambda, \boldsymbol{\Sigma}_{\boldsymbol{\beta}}) &= \boldsymbol{E}(y - \boldsymbol{x}^\top \hat{\boldsymbol{\beta}}_\Lambda)^2 \\ &= \boldsymbol{E}(\hat{\boldsymbol{\beta}}_\Lambda - \boldsymbol{\beta})^\top \boldsymbol{\Sigma}_{\boldsymbol{X}} (\hat{\boldsymbol{\beta}}_\Lambda - \boldsymbol{\beta}).\end{aligned}$$

Optimal shaping matrix

$$\Lambda^* = \arg \min_{\Lambda' \in \boldsymbol{S}^d_{++}} \mathrm{risk}(\Lambda', \boldsymbol{\Sigma}_{\boldsymbol{\beta}})$$

Error metric for asymptotic case

Risk function

$$\begin{aligned}\text{risk}(\Lambda, \boldsymbol{\Sigma}_{\boldsymbol{\beta}}) &= \boldsymbol{E}(y - \boldsymbol{x}^\top \hat{\boldsymbol{\beta}}_\Lambda)^2 \\ &= \boldsymbol{E}(\hat{\boldsymbol{\beta}}_\Lambda - \boldsymbol{\beta})^\top \boldsymbol{\Sigma}_{\boldsymbol{X}} (\hat{\boldsymbol{\beta}}_\Lambda - \boldsymbol{\beta}).\end{aligned}$$

Optimal shaping matrix

$$\Lambda^* = \arg \min_{\Lambda' \in \boldsymbol{S}^d_{++}} \text{risk}(\Lambda', \boldsymbol{\Sigma}_{\boldsymbol{\beta}})$$

As we do not know $\boldsymbol{\Sigma}_{\boldsymbol{\beta}}$, define

$$\Lambda = \arg \min_{\Lambda' \in \boldsymbol{S}^d_{++}} \text{risk}(\Lambda', \hat{\boldsymbol{\Sigma}}_{\boldsymbol{\beta}})$$

Error metric for asymptotic case

Risk function

$$\begin{aligned}\text{risk}(\Lambda, \boldsymbol{\Sigma}_{\boldsymbol{\beta}}) &= \boldsymbol{E}(y - \boldsymbol{x}^\top \hat{\boldsymbol{\beta}}_\Lambda)^2 \\ &= \boldsymbol{E}(\hat{\boldsymbol{\beta}}_\Lambda - \boldsymbol{\beta})^\top \boldsymbol{\Sigma}_{\boldsymbol{X}} (\hat{\boldsymbol{\beta}}_\Lambda - \boldsymbol{\beta}).\end{aligned}$$

Optimal shaping matrix

$$\Lambda^* = \arg \min_{\Lambda' \in \boldsymbol{S}^d_{++}} \text{risk}(\Lambda', \boldsymbol{\Sigma}_{\boldsymbol{\beta}})$$

As we do not know $\boldsymbol{\Sigma}_{\boldsymbol{\beta}}$, define

$$\Lambda = \arg \min_{\Lambda' \in \boldsymbol{S}^d_{++}} \text{risk}(\Lambda', \hat{\boldsymbol{\Sigma}}_{\boldsymbol{\beta}})$$

Asymptotic: Let $n_{\text{fs}}, d \to \infty$ and n_{fs}/d be fixed.

Computing shaping matrix

Asymptotic: Let $n_{\mathrm{fs}}, d \to \infty$ and n_{fs}/d be fixed.
Let $\mathbf{\Sigma}_{\boldsymbol{X}} = \boldsymbol{I}$ and $\mathbf{\Sigma}_{\boldsymbol{\beta}}$ be diagonal. Let ξ solve

$n_{\mathrm{fs}} = \sum_{i=1}^{d} \left(1 + (\xi \mathbf{\Sigma}_{\boldsymbol{X}\,i})^{-1}\right)^{-1}$.
Define $\boldsymbol{\theta} \in \mathbb{R}^d$ to be $\boldsymbol{\theta}_i = \frac{\xi \Lambda_i^2}{1+\xi\Lambda_i^2}$, and the risk is

$$\mathrm{risk}(\Lambda, \hat{\mathbf{\Sigma}}_{\boldsymbol{\beta}}) = \frac{1}{n_{\mathrm{fs}} - \|\boldsymbol{\theta}\|^2} \left(\frac{n_{\mathrm{fs}}}{d} \sum_{i=1}^{d} (1 - \boldsymbol{\theta}_i)^2 \hat{\mathbf{\Sigma}}_{\boldsymbol{\beta} i} + \|\boldsymbol{\theta}\|^2 \sigma_\varepsilon^2 \right).$$

We denote the right hand side as $f(\boldsymbol{\theta}; \hat{\mathbf{\Sigma}}_{\boldsymbol{\beta}})$.

Computing shaping matrix

Asymptotic: Let $n_{\mathrm{fs}}, d \to \infty$ and n_{fs}/d be fixed.
Let $\boldsymbol{\Sigma}_{\boldsymbol{X}} = \boldsymbol{I}$ and $\boldsymbol{\Sigma}_{\boldsymbol{\beta}}$ be diagonal. Let ξ solve

$n_{\mathrm{fs}} = \sum_{i=1}^{d} \left(1 + (\xi \boldsymbol{\Sigma}_{\boldsymbol{X}\,i})^{-1}\right)^{-1}$.
Define $\boldsymbol{\theta} \in \mathbb{R}^d$ to be $\boldsymbol{\theta}_i = \frac{\xi \Lambda_i^2}{1+\xi\Lambda_i^2}$, and the risk is

$$\mathrm{risk}(\Lambda, \hat{\boldsymbol{\Sigma}}_{\boldsymbol{\beta}}) = \frac{1}{n_{\mathrm{fs}} - \|\boldsymbol{\theta}\|^2} \left(\frac{n_{\mathrm{fs}}}{d} \sum_{i=1}^{d} (1 - \boldsymbol{\theta}_i)^2 \hat{\boldsymbol{\Sigma}}_{\boldsymbol{\beta} i} + \|\boldsymbol{\theta}\|^2 \sigma_{\varepsilon}^2 \right).$$

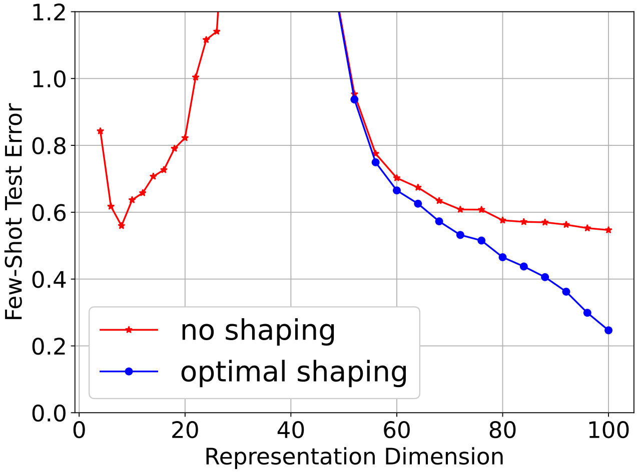
We denote the right hand side as $f(\boldsymbol{\theta}; \hat{\boldsymbol{\Sigma}}_{\boldsymbol{\beta}})$.

Computation of optimal representation:

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \; f(\boldsymbol{\theta}; \hat{\boldsymbol{\Sigma}}_{\boldsymbol{\beta}}), \text{ s.t. } \underline{\theta} \le \boldsymbol{\theta} < 1, \sum_{i=1}^{d} \boldsymbol{\theta}_i = n_{\mathrm{fs}}.$$

$$\Lambda_i^* = ((1/\boldsymbol{\theta}_i^* - 1)\xi)^{-2}$$

Double descent



Error of meta-learning

Suppose $\mathcal{E}$ is the error of representation learning.

$$\mathsf{risk}(\Lambda, \mathbf{\Sigma}_{\boldsymbol{\beta}}) \leq \mathsf{risk}(\Lambda^*, \mathbf{\Sigma}_{\boldsymbol{\beta}}) + \mathcal{O}\left(\frac{n_{\mathrm{fs}}^2 \cdot \mathcal{E}}{(d - n_{\mathrm{fs}})(2n_{\mathrm{fs}} - d\underline{\theta})\underline{\theta}}\right)$$

Error of meta-learning

Suppose $\mathcal{E}$ is the error of representation learning.

$$\mathrm{risk}(\Lambda, \mathbf{\Sigma}_{\boldsymbol{\beta}}) \leq \mathrm{risk}(\Lambda^*, \mathbf{\Sigma}_{\boldsymbol{\beta}}) + \mathcal{O}\left(\frac{n_{\mathrm{fs}}^2 \cdot \mathcal{E}}{(d - n_{\mathrm{fs}})(2n_{\mathrm{fs}} - d\underline{\theta})\underline{\theta}}\right)$$

We have presented $\Lambda \in \mathbb{R}^{d \times d}$. We can similarly define a $\mathbb{R}^{R \times d}$ representation Λ_R for arbitrary $R > n_{\mathrm{fs}}$ by projecting onto a subspace.

Error of meta-learning

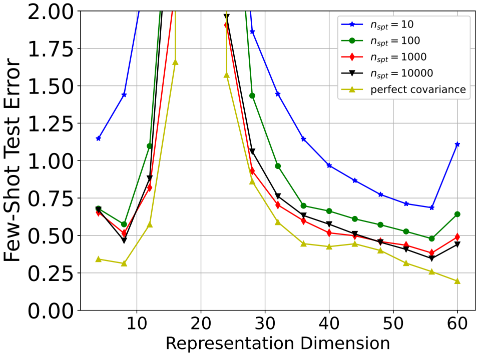
Suppose $\mathcal{E}$ is the error of representation learning.

$$\mathrm{risk}(\Lambda, \boldsymbol{\Sigma}_{\boldsymbol{\beta}}) \leq \mathrm{risk}(\Lambda^*, \boldsymbol{\Sigma}_{\boldsymbol{\beta}}) + \mathcal{O}\left(\frac{n_{\mathrm{fs}}^2 \cdot \mathcal{E}}{(d - n_{\mathrm{fs}})(2n_{\mathrm{fs}} - d\underline{\theta})\underline{\theta}}\right)$$

We have presented $\Lambda \in \mathbb{R}^{d \times d}$. We can similarly define a $\mathbb{R}^{R \times d}$ representation Λ_R for arbitrary $R > n_{\mathrm{fs}}$ by projecting onto a subspace.

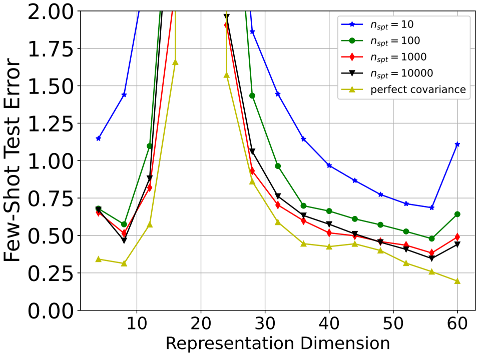
Tradeoff: when R increases, $\mathrm{risk}(\Lambda_R^*, \boldsymbol{\Sigma}_{\boldsymbol{\beta}})$ decreases, $\mathcal{E}$ increases.

Empirical observation



We plot the error of few-shot learning versus varying dimension of Λ. Different curves correspond to different sample size for rep learning.

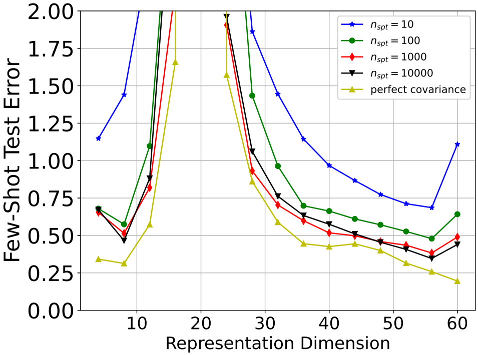
Empirical observation



We plot the error of few-shot learning versus varying dimension of Λ. Different curves correspond to different sample size for rep learning.

When rep learning sample size $= \infty$, $\hat{\boldsymbol{\Sigma}}_{\boldsymbol{\beta}} = \boldsymbol{\Sigma}_{\boldsymbol{\beta}}$, smallest error at $R = d$.

Empirical observation



We plot the error of few-shot learning versus varying dimension of Λ. Different curves correspond to different sample size for rep learning.

When rep learning sample size $= \infty$, $\hat{\mathbf{\Sigma}}_{\boldsymbol{\beta}} = \mathbf{\Sigma}_{\boldsymbol{\beta}}$, smallest error at $R = d$. Finite sample, $\hat{\mathbf{\Sigma}}_{\boldsymbol{\beta}} \neq \mathbf{\Sigma}_{\boldsymbol{\beta}}$, smallest error when R is slightly smaller than d.

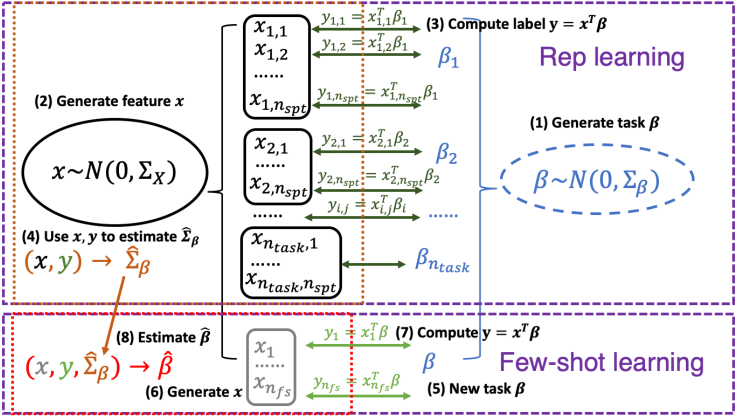
Overview

Representation learning - Linear

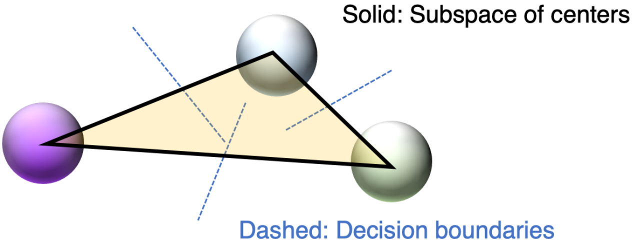
Few-shot learning - Linear

Meta learning - Nonlinear

Meta-learning - Linear



Motivating example: Multi-class classification



$$\dim\big(\mathrm{span}(\mathrm{centers})\big) < \dim(\mathrm{space})$$

Meta-learning - Nonlinear - Dataset

- Fix a matrix $\boldsymbol{W} \in \mathbb{R}^{r \times d}$ satisfying $\boldsymbol{W}\boldsymbol{W}^\top = \boldsymbol{I}$.
- The i-th task is associated with function $f^i : \mathbb{R}^r \to \mathbb{R}$.
- Given input $\boldsymbol{x} \in \mathbb{R}^d$, the label y is distributed as $p_i(y|\boldsymbol{x}) = p_i(y|\boldsymbol{W}\boldsymbol{x})$ and the expectation satisfies $\boldsymbol{E}(y) = f^i(\boldsymbol{W}\boldsymbol{x})$.

In words, the label depends on the relevant features induced by $\boldsymbol{W}$.

Meta-learning - Nonlinear - Dataset

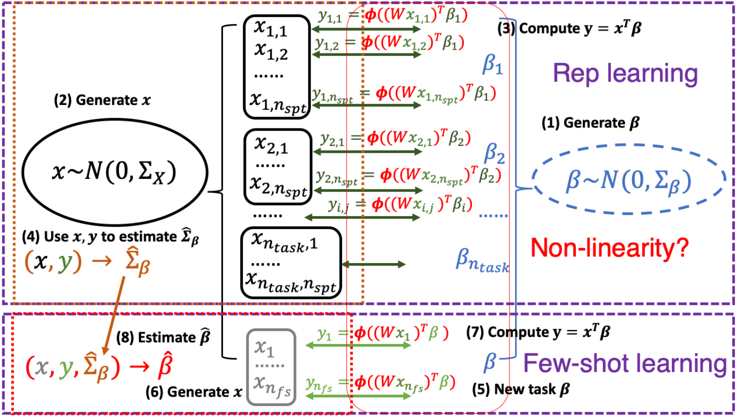
- Fix a matrix $\boldsymbol{W} \in \mathbb{R}^{r \times d}$ satisfying $\boldsymbol{W}\boldsymbol{W}^\top = \boldsymbol{I}$.
- The i-th task is associated with function $f^i : \mathbb{R}^r \to \mathbb{R}$.
- Given input $\boldsymbol{x} \in \mathbb{R}^d$, the label y is distributed as $p_i(y|\boldsymbol{x}) = p_i(y|\boldsymbol{W}\boldsymbol{x})$ and the expectation satisfies $\boldsymbol{E}(y) = f^i(\boldsymbol{W}\boldsymbol{x})$.

In words, the label depends on the relevant features induced by $\boldsymbol{W}$.

Example: Generalized linear models (GLM), which include logistic/linear regression, can be modeled by choosing f^i to be parameterized by a vector $\boldsymbol{\beta}_i \in \mathbb{R}^r$ and a link function $\phi : \mathbb{R} \to \mathbb{R}$ as $f^i(\boldsymbol{W}\boldsymbol{x}_{ij}) := \phi((\boldsymbol{W}\boldsymbol{x}_{ij})^\top \boldsymbol{\beta}_i)$.

– logistic regression, multi-class classification, etc.

Meta-learning - Nonlinear



Representation learning

Moment estimator of covariance.

$$\hat{\boldsymbol{M}} = \sum_{i=1}^{k} \frac{2}{n_i^2} \left[(\sum_{j=1}^{n_i/2} y_{ij}\boldsymbol{x}_{ij})(\sum_{j=n_i/2+1}^{n_i} y_{ij}\boldsymbol{x}_{ij})^\top \right.$$
$$\left. +(\sum_{j=n_i/2+1}^{n_i} y_{ij}\boldsymbol{x}_{ij})(\sum_{i=1}^{n_i/2} y_{ij}\boldsymbol{x}_{ij})^\top \right]$$

Define

$$\boldsymbol{h}^i(\boldsymbol{W}) : \mathbb{R}^{r\times d} \to \mathbb{R}^d = \boldsymbol{E}_{\boldsymbol{x}}[f^i(\boldsymbol{W}\boldsymbol{x})\boldsymbol{x}]$$
$$\boldsymbol{M} := \boldsymbol{W}^\top \boldsymbol{W} \left(\frac{1}{k} \sum_{i=1}^{k} \boldsymbol{h}^i(\boldsymbol{W})(\boldsymbol{h}^i(\boldsymbol{W}))^\top \right) \boldsymbol{W}^\top \boldsymbol{W}.$$

Representation learning - Result

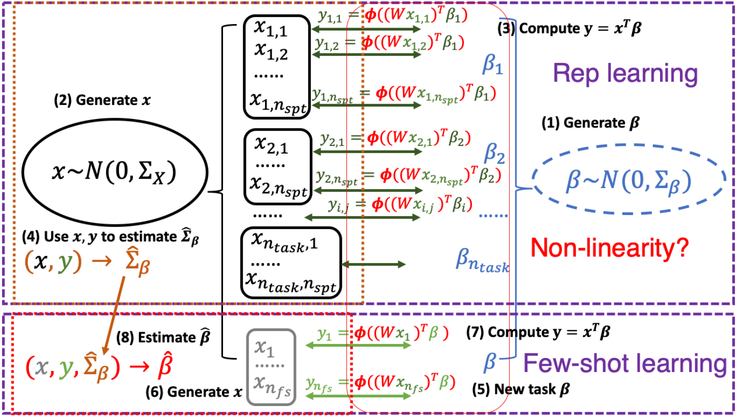
Suppose $y\boldsymbol{x}$ is a subGaussian random vector with covariance upper bounded by $\|\mathrm{Cov}(y\boldsymbol{x})\| \leq \sigma^2$. (These conditions hold when $|f^i(\boldsymbol{x})| < \sigma$.) Let $\epsilon \in (0,1)$. Then there exists a constant $c > 0$ such that

$$k \gtrsim \frac{cd}{\min\{n_i\}\epsilon^2} \Rightarrow \|\hat{\boldsymbol{M}} - \boldsymbol{M}\| \leq \epsilon\sigma^2$$

with probability at least $1-\delta$. If $\lambda_r(\boldsymbol{M}) > \epsilon\sigma^2$, then for some orthonormal matrix $\boldsymbol{Q} \in \mathbb{R}^{r\times r}$,

$$\|\hat{\boldsymbol{W}} - \boldsymbol{Q}\boldsymbol{W}\| \leq \epsilon\sigma^2(\lambda_r(\boldsymbol{M}) - \epsilon\sigma^2)^{-1}.$$

Meta-learning - Nonlinear



Few-shot learning - Metric

Let $\mathscr{P}_{x,y}$ be the joint distribution of $\boldsymbol{x}, y$. We introduce population risk $\mathcal{L}$ and empirical risk $\mathcal{L}_e$ based on any single loss function between model prediction and true label.

$$\mathcal{L}(f; \boldsymbol{P}) = \boldsymbol{E}_{\mathscr{P}_{x,y}} \mathrm{loss}(f(\boldsymbol{P}\boldsymbol{x}), y)$$
$$\mathcal{L}_e(f; \boldsymbol{P}) = \frac{1}{n} \sum_{i=1}^{n} \mathrm{loss}(f(\boldsymbol{P}\boldsymbol{x}_i), y_i).$$

We make the following assumption on the population risk.
1. $\mathcal{L}$ is L Lipschitz in $\boldsymbol{P}\boldsymbol{x}$. 2. $\min_{\boldsymbol{P}} \mathcal{L}(f; \boldsymbol{P}) = \mathcal{L}(f; \boldsymbol{W})$.

Few-shot learning - Metric

Let $\mathscr{P}_{x,y}$ be the joint distribution of $\boldsymbol{x}, y$. We introduce population risk $\mathcal{L}$ and empirical risk $\mathcal{L}_e$ based on any single loss function between model prediction and true label.

$$\mathcal{L}(f;\boldsymbol{P}) = \boldsymbol{E}_{\mathscr{P}_{x,y}} \mathrm{loss}(f(\boldsymbol{Px}), y)$$
$$\mathcal{L}_e(f;\boldsymbol{P}) = \frac{1}{n}\sum_{i=1}^{n} \mathrm{loss}(f(\boldsymbol{Px}_i), y_i).$$

We make the following assumption on the population risk.
1. $\mathcal{L}$ is L Lipschitz in $\boldsymbol{Px}$. 2. $\min_{\boldsymbol{P}} \mathcal{L}(f;\boldsymbol{P}) = \mathcal{L}(f;\boldsymbol{W})$.

Example: Suppose there are n samples, f is an L-Lipschitz function with range in $(0,1)$. The cross entropy function satisfies the assumptions.

$$\mathcal{L}(f;\boldsymbol{P}) = -\boldsymbol{E}_{\mathscr{P}_{x,y}}(y \log f(\boldsymbol{Px}) + (1-y)\log(1-f(\boldsymbol{Px})))$$
$$\mathcal{L}_e(f;\boldsymbol{P}) = -\frac{1}{n}\sum_{i=1}^{n}(y_i \log(f(\boldsymbol{Px}_i)) + (1-y_i)\log(1-f(\boldsymbol{Px}_i))).$$

Few-shot learning

In the few-shot learning phase, suppose $\boldsymbol{x}, y \sim \mathscr{P}_{x,y}$ satisfy $\boldsymbol{E}[y \mid \boldsymbol{x}] = f^*(\boldsymbol{W}\boldsymbol{x})$. Let $\mathcal{F}$ be a family of functions as the search space for few-shot learning model. We search for the solution

$$\hat{f}_e = \operatorname{argmin}_{f \in \mathcal{F}} \mathcal{L}_e(f; \hat{\boldsymbol{W}})$$

Few-shot learning - Result

Suppose we have n i.i.d. examples with ground-truth model $f^*(\boldsymbol{x}) = \phi((\boldsymbol{W}\boldsymbol{x})^\top \theta^*)$ where[1] $\|\theta^*\| \leq a$. Let $\mathcal{F}$ be the family of functions of $\boldsymbol{x}$ expressed as $\{\phi((\hat{\boldsymbol{W}}\boldsymbol{x})^\top \theta) : \|\theta\| \leq a\}$, we solve for

$$\hat{f}_e = \mathrm{argmin}_{f \in \mathcal{F}}\, \mathcal{L}_e(f; \hat{\boldsymbol{W}})$$

There exist constants $c > 1$, $\delta \in (0, 1)$, with probability at least $1 - n^{-c+1} - \delta$,

$$\begin{aligned} &\mathcal{L}(\hat{f}_e; \hat{\boldsymbol{W}}) - \mathcal{L}(f^*; \boldsymbol{W}) \\ &\leq \frac{caL(\sqrt{r} + \log(n))(1 + \sqrt{\log(1/\delta)})}{\sqrt{n}} + L\sqrt{r}\|\hat{\boldsymbol{W}} - \boldsymbol{W}\|. \end{aligned}$$

[1] bounded norm for Rademacher complexity analysis.