

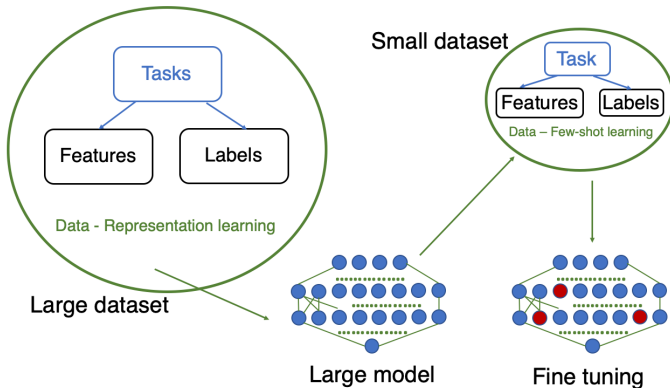
# Subspace Based Meta-Learning

Yue Sun  
University of Washington

Joint work with:  
Halil Ibrahim Gulluk (Bogazici University), 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:**  $\beta \sim \mathcal{N}(0, \Sigma_\beta)$ ,  $\Sigma_\beta$  approx low rank, **Feature:**  $\mathbf{x} \sim \mathcal{N}(0, \Sigma_{\mathbf{x}})$ ,

**Noise:**  $\varepsilon \sim \mathcal{N}(0, \sigma_\varepsilon)$ , **Label:**  $y = \mathbf{x}^\top \beta + \varepsilon$ .

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

# Meta learning

Task, feature in  $\mathbb{R}^d$ , label in  $\mathbb{R}$ .

**Task:**  $\beta \sim \mathcal{N}(0, \Sigma_\beta)$ ,  $\Sigma_\beta$  **approx low rank**, **Feature:**  $\mathbf{x} \sim \mathcal{N}(0, \Sigma_{\mathbf{x}})$ ,

**Noise:**  $\varepsilon \sim \mathcal{N}(0, \sigma_\varepsilon)$ , **Label:**  $y = \mathbf{x}^\top \beta + \varepsilon$ .

- ▶ **Two steps:** Representation learning, Few-shot learning
- ▶ **Rep learning:** Sample  $\beta_1, \dots, \beta_{n_{\text{task}}}$ . For  $i \in [n_{\text{task}}]$ , Sample  $\mathbf{x}_{i,1}, \dots, \mathbf{x}_{i,n_{\text{spt}}}$ . Evaluate  $y$ . Use  $\mathbf{x}, y$  to estimate  $\Sigma_\beta$ .

# Meta learning

Task, feature in  $\mathbb{R}^d$ , label in  $\mathbb{R}$ .

**Task:**  $\beta \sim \mathcal{N}(0, \Sigma_\beta)$ ,  $\Sigma_\beta$  **approx low rank**, **Feature:**  $\mathbf{x} \sim \mathcal{N}(0, \Sigma_{\mathbf{x}})$ ,

**Noise:**  $\varepsilon \sim \mathcal{N}(0, \sigma_\varepsilon)$ , **Label:**  $y = \mathbf{x}^\top \beta + \varepsilon$ .

- ▶ **Two steps:** Representation learning, Few-shot learning
- ▶ **Rep learning:** Sample  $\beta_1, \dots, \beta_{n_{\text{task}}}$ . For  $i \in [n_{\text{task}}]$ , Sample  $\mathbf{x}_{i,1}, \dots, \mathbf{x}_{i,n_{\text{spt}}}$ . Evaluate  $y$ . Use  $\mathbf{x}, y$  to estimate  $\Sigma_\beta$ .
- ▶ **Few-shot learning:** Sample  $\beta, \mathbf{x}_1, \dots, \mathbf{x}_{n_{\text{spt}}}$ , evaluate  $y$ . Use  $\mathbf{x}, y$  and estimate  $\beta$  in principal subspace of  $\Sigma_\beta$ .

# Meta learning

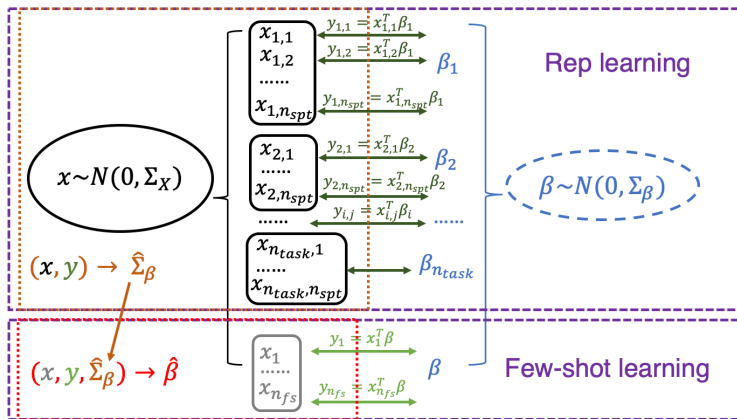
Task, feature in  $\mathbb{R}^d$ , label in  $\mathbb{R}$ .

**Task:**  $\beta \sim \mathcal{N}(0, \Sigma_\beta)$ ,  $\Sigma_\beta$  **approx low rank**, **Feature:**  $\mathbf{x} \sim \mathcal{N}(0, \Sigma_{\mathbf{x}})$ ,

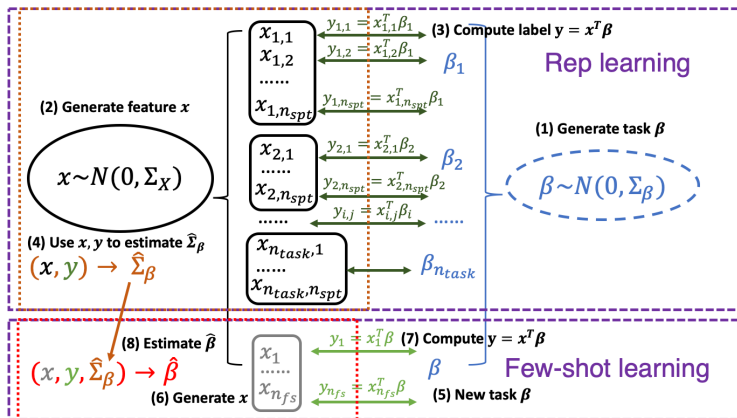
**Noise:**  $\varepsilon \sim \mathcal{N}(0, \sigma_\varepsilon)$ , **Label:**  $y = \mathbf{x}^\top \beta + \varepsilon$ .

- ▶ **Two steps:** Representation learning, Few-shot learning
- ▶ **Rep learning:** Sample  $\beta_1, \dots, \beta_{n_{\text{task}}}$ . For  $i \in [n_{\text{task}}]$ , Sample  $\mathbf{x}_{i,1}, \dots, \mathbf{x}_{i,n_{\text{spt}}}$ . Evaluate  $y$ . Use  $\mathbf{x}, y$  to estimate  $\Sigma_\beta$ .
- ▶ **Few-shot learning:** Sample  $\beta, \mathbf{x}_1, \dots, \mathbf{x}_{n_{\text{spt}}}$ , evaluate  $y$ . Use  $\mathbf{x}, y$  and estimate  $\beta$  in principal subspace of  $\Sigma_\beta$ .
- ▶ **Few-shot learning:** Sample  $\beta, \mathbf{x}_1, \dots, \mathbf{x}_{n_{\text{fs}}}$ , evaluate  $y$ . Use  $\mathbf{x}, y$  and a shaping matrix as a function of  $\hat{\Sigma}_\beta$  to estimate  $\beta$ .

# 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 $\Sigma_\beta$

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

**Rep learning:** Sample  $\beta_1, \dots, \beta_{n_{\text{task}}}$ . For  $i \in [n_{\text{task}}]$ , Sample  $\mathbf{x}_{i,1}, \dots, \mathbf{x}_{i,n_{\text{spt}}}$ . Evaluate  $y$ . Use  $x, y$  to estimate  $\Sigma_\beta$ .

## Rep learning - learn $\Sigma_\beta$

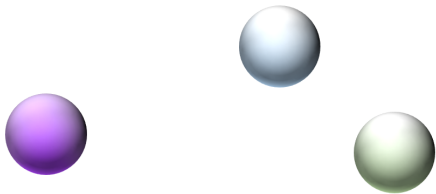
**Task:**  $\beta \sim \mathcal{N}(0, \Sigma_\beta)$ , **Feature:**  $\mathbf{x} \sim \mathcal{N}(0, \Sigma_\mathbf{x})$ , **Label:**  $y = \mathbf{x}^\top \beta$ .  
Suppose  $\text{rank}(\Sigma_\mathbf{x}) = r_f$ ,  $\text{rank}(\Sigma_\beta) = r_t$ . Suppose the principal subspaces of  $\Sigma_\mathbf{x}$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

**Rep learning:** Sample  $\beta_1, \dots, \beta_{n_{\text{task}}}$ . For  $i \in [n_{\text{task}}]$ , Sample  $\mathbf{x}_{i,1}, \dots, \mathbf{x}_{i,n_{\text{spt}}}$ . Evaluate  $y$ . Use  $x, y$  to estimate  $\Sigma_\beta$ .

Interested in **feature-task alignment**.

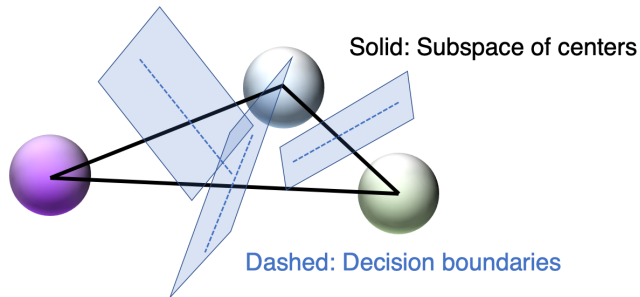
E.g.,  $\Sigma_\beta = \text{diag}(\mathbf{I}_{r_t}, 0)$ ,  $\Sigma_\mathbf{x} = \text{diag}(\mathbf{I}_{r_f}, \mathbf{I}_{d-r_f})$ .

## Motivating example: Multi-class classification

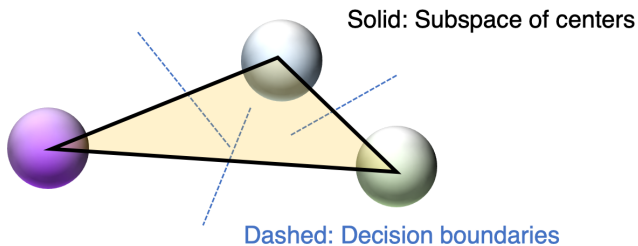


Motivation: classification of Gaussian mixture

## Motivating example: Multi-class classification



## Motivating example: Multi-class classification



$$\dim(\text{span}(\text{centers})) < \dim(\text{space})$$

## Naive case: estimating $\Sigma_{\mathbf{X}}$ is enough

1.  $\Sigma_{\beta} = \Sigma_{\mathbf{X}}$ .
2.  $\text{span}(\Sigma_{\beta}) = \text{span}(\Sigma_{\mathbf{X}})$ .
3.  $\text{span}(\Sigma_{\beta}) \subset \text{span}(\Sigma_{\mathbf{X}})$  but we are satisfied with  $\text{span}(\Sigma_{\mathbf{X}})$ .



## Naive case: estimating $\Sigma_{\mathbf{X}}$ is enough

1.  $\Sigma_{\beta} = \Sigma_{\mathbf{X}}$ .
2.  $\text{span}(\Sigma_{\beta}) = \text{span}(\Sigma_{\mathbf{X}})$ .
3.  $\text{span}(\Sigma_{\beta}) \subset \text{span}(\Sigma_{\mathbf{X}})$  but we are satisfied with  $\text{span}(\Sigma_{\mathbf{X}})$ .

**MoM-F estimator:**

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

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

## General case: MoM estimator

When  $n_{\text{spt}}$  is small.

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

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

## General case: MoM estimator

When  $n_{\text{spt}}$  is small.

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

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

Mean:

$$\mathbf{Q} = 2 \Sigma_{\mathbf{X}} \Sigma_{\beta} \Sigma_{\mathbf{X}} + \text{tr}(\Sigma_{\beta} \Sigma_{\mathbf{X}}) \Sigma_{\mathbf{X}}.$$

$$\mathbf{M} = \Sigma_{\mathbf{X}} \Sigma_{\beta} \Sigma_{\mathbf{X}}.$$

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

## General case: MoM-TA estimator

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

$$\hat{\mathbf{B}} = [\hat{\mathbf{b}}_1, \dots, \hat{\mathbf{b}}_{n_{\text{task}}}],$$

$$\hat{\mathbf{G}} = n_{\text{task}}^{-1} \hat{\mathbf{B}} \hat{\mathbf{B}}^\top.$$

## General case: MoM-TA estimator

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

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

Mean:

$$\mathbf{G} = \Sigma_{\mathbf{X}} \Sigma_{\beta} \Sigma_{\mathbf{X}} + n_{\text{spt}}^{-1} (\Sigma_{\mathbf{X}} \Sigma_{\beta} \Sigma_{\mathbf{X}} + \text{tr}(\Sigma_{\beta} \Sigma_{\mathbf{X}}) \Sigma_{\mathbf{X}})$$

Sample complexity:

1. Generally  $\mathcal{O}(r_f r_t^2)$ .
2.  $\mathcal{O}(r_f r_t)$  when  $n_{\text{spt}} \geq r_t$ .

## Rep learning - learn $\Sigma_\beta$

**Task:**  $\beta \sim \mathcal{N}(0, \Sigma_\beta)$ , **Feature:**  $\mathbf{x} \sim \mathcal{N}(0, \Sigma_\mathbf{x})$ , **Label:**  $y = \mathbf{x}^\top \beta$ .

Suppose  $\text{rank}(\Sigma_\mathbf{x}) = r_f$ ,  $\text{rank}(\Sigma_\beta) = r_t$ . Suppose the principal subspaces of  $\Sigma_\mathbf{x}$  and  $\Sigma_\beta$  align.

**Rep learning:** Sample  $\beta_1, \dots, \beta_{n_{\text{task}}}$ . For  $i \in [n_{\text{task}}]$ , Sample  $\mathbf{x}_{i,1}, \dots, \mathbf{x}_{i,n_{\text{spt}}}$ . Evaluate  $y$ . Use  $\mathbf{x}, y$  to estimate  $\Sigma_\beta$ .

**Estimators:** MoM, MoM-TA, MoM-F.

MoM:  $\sum_{i,j} y_{ij}^2 \mathbf{x}_{ij} \mathbf{x}_{ij}^\top$ .

MoM-TA: Let  $\hat{\mathbf{b}}_i = \sum_{j=1}^{n_{\text{spt}}} y_{ij} \mathbf{x}_{ij}$ .  $\hat{\mathbf{B}} = [\hat{\mathbf{b}}_1, \dots, \hat{\mathbf{b}}_{n_{\text{task}}}]$ . Need  $n_{\text{spt}} \geq r_t$ .

MoM-F:  $\sum_{i,j} \mathbf{x}_{ij} \mathbf{x}_{ij}^\top$ .

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

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

## Tradeoff between $n_{\text{spt}}$ and $n_{\text{task}}$

$n_{\text{spt}}$ : Sample per task

$n_{\text{task}}$ : Number of tasks

$n_{\text{tot}}$ : Total samples  $= n_{\text{spt}} n_{\text{task}}$ .

**Question:** Fix  $n_{\text{tot}}$  and change  $n_{\text{spt}}, n_{\text{task}}$ .

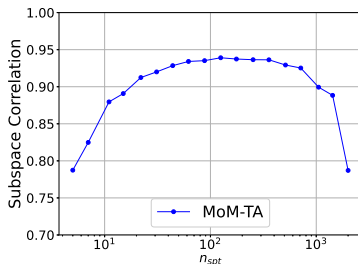
# Tradeoff between $n_{\text{spt}}$ and $n_{\text{task}}$

$n_{\text{spt}}$ : Sample per task

$n_{\text{task}}$ : Number of tasks

$n_{\text{tot}}$ : Total samples =  $n_{\text{spt}} n_{\text{task}}$ .

**Question:** Fix  $n_{\text{tot}}$  and change  $n_{\text{spt}}$ ,  $n_{\text{task}}$ .





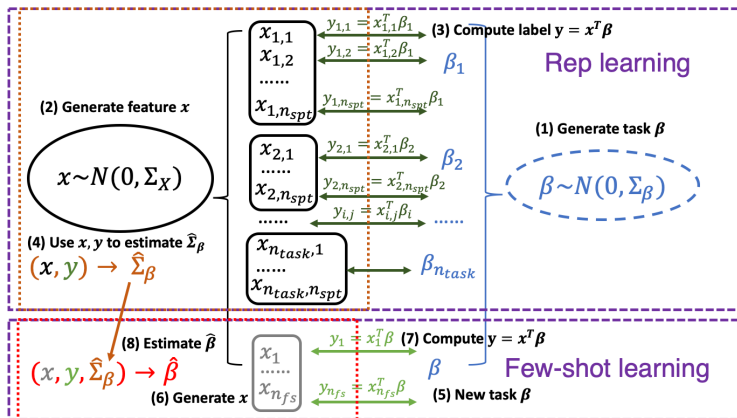
# Overview

Representation learning - Linear

Few-shot learning - Linear

Meta learning - Nonlinear

# Meta learning



## Few-shot learning - learn $\beta$

**Task:**  $\beta \sim \mathcal{N}(0, \Sigma_\beta)$ , **Feature:**  $\mathbf{x} \sim \mathcal{N}(0, \Sigma_{\mathbf{x}})$ , **Label:**  $y = \mathbf{x}^\top \beta$ .

Suppose  $\text{rank}(\Sigma_{\mathbf{x}}) = r_f$ ,  $\text{rank}(\Sigma_\beta) = r_t$ .

**Few-shot learning:** Sample  $\beta$ ,  $\mathbf{x}_1, \dots, \mathbf{x}_{n_{\text{fs}}}$ , evaluate  $y$ . Use  $\mathbf{x}, y$  and a shaping matrix as a function of  $\hat{\Sigma}_\beta$  to estimate  $\beta$ .

## Few-shot learning - learn $\beta$

**Task:**  $\beta \sim \mathcal{N}(0, \Sigma_\beta)$ , **Feature:**  $\mathbf{x} \sim \mathcal{N}(0, \Sigma_{\mathbf{x}})$ , **Label:**  $y = \mathbf{x}^\top \beta$ .  
Suppose  $\text{rank}(\Sigma_{\mathbf{x}}) = r_f$ ,  $\text{rank}(\Sigma_\beta) = r_t$ .

**Few-shot learning:** Sample  $\beta$ ,  $\mathbf{x}_1, \dots, \mathbf{x}_{n_{\text{fs}}}$ , evaluate  $y$ . Use  $\mathbf{x}, y$  and a shaping matrix as a function of  $\hat{\Sigma}_\beta$  to estimate  $\beta$ .

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

## Few-shot learning - learn $\beta$

**Task:**  $\beta \sim \mathcal{N}(0, \Sigma_\beta)$ , **Feature:**  $\mathbf{x} \sim \mathcal{N}(0, \Sigma_{\mathbf{x}})$ , **Label:**  $y = \mathbf{x}^\top \beta$ .  
Suppose  $\text{rank}(\Sigma_{\mathbf{x}}) = r_f$ ,  $\text{rank}(\Sigma_\beta) = r_t$ .

**Few-shot learning:** Sample  $\beta$ ,  $\mathbf{x}_1, \dots, \mathbf{x}_{n_{\text{fs}}}$ , evaluate  $y$ . Use  $\mathbf{x}, y$  and a shaping matrix as a function of  $\hat{\Sigma}_\beta$  to estimate  $\beta$ .

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

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

## How does shaping matrix work

**Task:**  $\beta \sim \mathcal{N}(0, \Sigma_\beta)$ , **Feature:**  $\mathbf{x} \sim \mathcal{N}(0, \Sigma_{\mathbf{x}})$ , **Label:**  $y = \mathbf{x}^\top \beta$ .

Suppose  $\text{rank}(\Sigma_{\mathbf{x}}) = r_f$ ,  $\text{rank}(\Sigma_\beta) = r_t$ .

**Few-shot learning:** Sample  $\beta$ ,  $\mathbf{x}_1, \dots, \mathbf{x}_{n_{\text{fs}}}$ , evaluate  $y$ . Use  $\mathbf{x}, y$  and a shaping matrix as a function of  $\hat{\Sigma}_\beta$  to estimate  $\beta$ .

**Min norm solution** with  $\Lambda$ .

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

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

## How does shaping matrix work

**Task:**  $\beta \sim \mathcal{N}(0, \Sigma_\beta)$ , **Feature:**  $\mathbf{x} \sim \mathcal{N}(0, \Sigma_{\mathbf{x}})$ , **Label:**  $y = \mathbf{x}^\top \beta$ .

Suppose  $\text{rank}(\Sigma_{\mathbf{x}}) = r_f$ ,  $\text{rank}(\Sigma_\beta) = r_t$ .

**Few-shot learning:** Sample  $\beta$ ,  $\mathbf{x}_1, \dots, \mathbf{x}_{n_{\text{fs}}}$ , evaluate  $y$ . Use  $\mathbf{x}, y$  and a shaping matrix as a function of  $\hat{\Sigma}_\beta$  to estimate  $\beta$ .

**Min norm solution** with  $\Lambda$ .

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

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

$$\hat{\beta}_\Lambda = \lim_{t \rightarrow 0} \arg \min_{\beta} \|\mathbf{X}^\top \beta - \mathbf{y}\|^2 + t \beta^\top \Lambda^{-2} \beta$$

## Error metric for asymptotic case

### Risk function

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



## Error metric for asymptotic case

### Risk function

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

### Optimal shaping matrix

$$\Lambda^* = \arg \min_{\Lambda' \in \mathbf{S}_{++}^d} \text{risk}(\Lambda', \Sigma_\beta)$$

## Error metric for asymptotic case

### Risk function

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

### Optimal shaping matrix

$$\Lambda^* = \arg \min_{\Lambda' \in \mathbf{S}_{++}^d} \text{risk}(\Lambda', \Sigma_\beta)$$

As we do not know  $\Sigma_\beta$ , define

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

## Error metric for asymptotic case

### Risk function

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

### Optimal shaping matrix

$$\Lambda^* = \arg \min_{\Lambda' \in \mathbf{S}_{++}^d} \text{risk}(\Lambda', \Sigma_\beta)$$

As we do not know  $\Sigma_\beta$ , define

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

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

## Computing shaping matrix

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

Let  $\Sigma_{\mathbf{X}} = \mathbf{I}$  and  $\Sigma_{\beta}$  be diagonal. Let  $\xi$  solve

$$n_{\text{fs}} = \sum_{i=1}^d \left(1 + (\xi \Sigma_{\mathbf{X}_i})^{-1}\right)^{-1}.$$

Define  $\boldsymbol{\theta} \in \mathbb{R}^d$  to be  $\theta_i = \frac{\xi \Lambda_i^2}{1 + \xi \Lambda_i^2}$ , and the risk is

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

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

## Computing shaping matrix

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

Let  $\Sigma_{\mathbf{X}} = \mathbf{I}$  and  $\Sigma_{\beta}$  be diagonal. Let  $\xi$  solve

$$n_{\text{fs}} = \sum_{i=1}^d (1 + (\xi \Sigma_{\mathbf{X}})_i)^{-1}.$$

Define  $\boldsymbol{\theta} \in \mathbb{R}^d$  to be  $\theta_i = \frac{\xi \Lambda_i^2}{1 + \xi \Lambda_i^2}$ , and the risk is

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

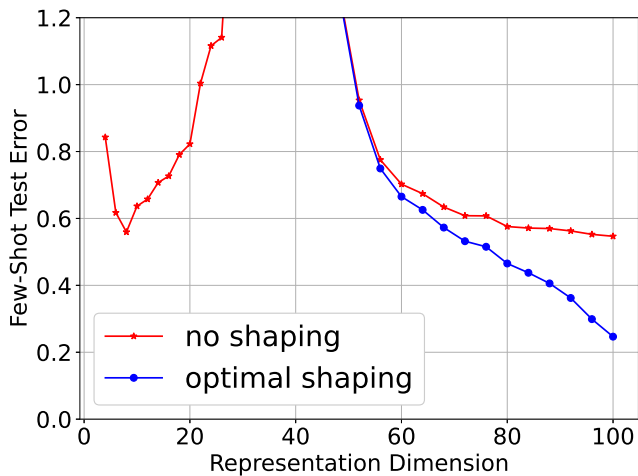
We denote the right hand side as  $f(\boldsymbol{\theta}; \hat{\Sigma}_{\beta})$ .

**Computation** of optimal representation:

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

$$\Lambda_i^* = ((1/\theta_i^* - 1)\xi)^{-2}$$

## Double descent



## Error of meta-learning

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

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

## Error of meta-learning

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

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

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



## Error of meta-learning

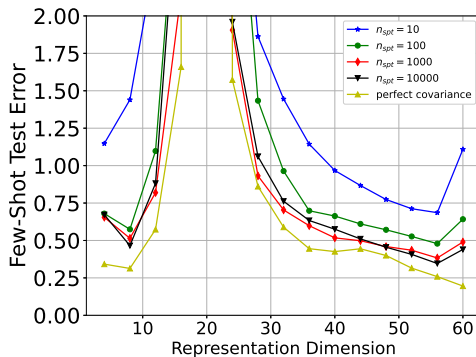
Suppose  $\mathcal{E}$  is the error of representation learning.

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

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

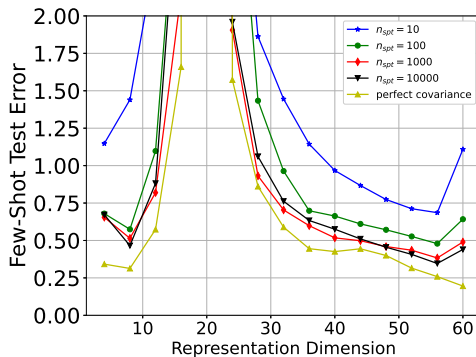
**Tradeoff:** when  $R$  increases,  $\text{risk}(\Lambda_R^*, \Sigma_\beta)$  decreases,  $\mathcal{E}$  increases.

## Empirical observation



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

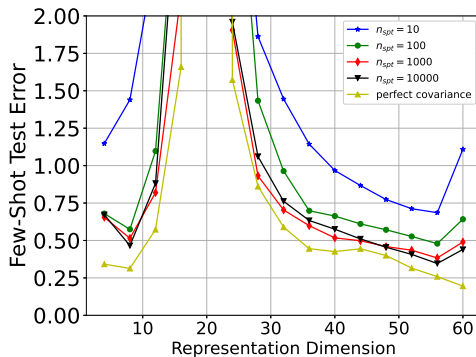
## Empirical observation



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

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

# Empirical observation



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

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

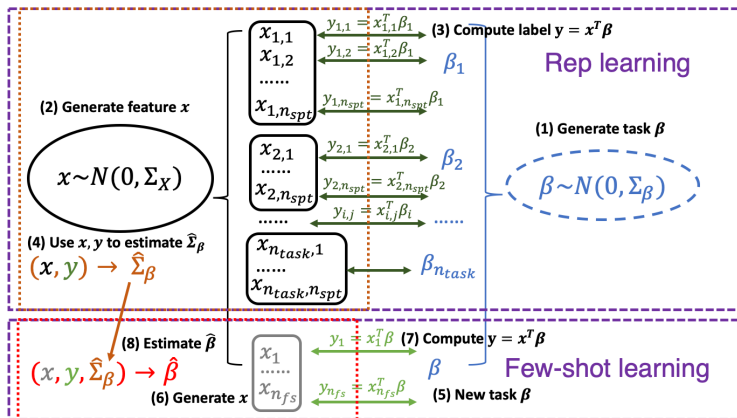
# Overview

Representation learning - Linear

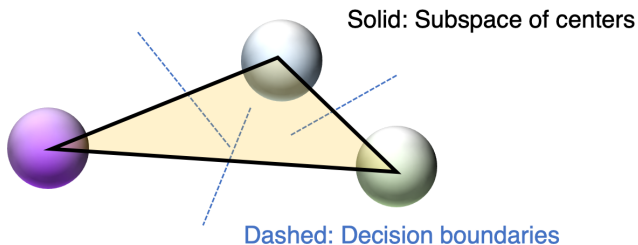
Few-shot learning - Linear

Meta learning - Nonlinear

# Meta-learning - Linear



## Motivating example: Multi-class classification



$$\dim(\text{span}(\text{centers})) < \dim(\text{space})$$

## Meta-learning - Nonlinear - Dataset

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

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



## Meta-learning - Nonlinear - Dataset

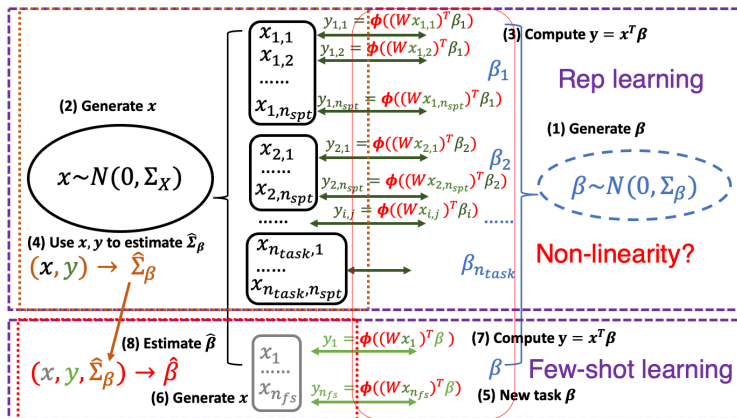
- ▶ Fix a matrix  $\mathbf{W} \in \mathbb{R}^{r \times d}$  satisfying  $\mathbf{W}\mathbf{W}^\top = \mathbf{I}$ .
- ▶ The  $i$ -th task is associated with function  $f^i : \mathbb{R}^r \rightarrow \mathbb{R}$ .
- ▶ Given input  $\mathbf{x} \in \mathbb{R}^d$ , the label  $y$  is distributed as  $p_i(y|\mathbf{x})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

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

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

– logistic regression, multi-class classification, etc.

# Meta-learning - Nonlinear



# Representation learning

**Moment estimator of covariance.**

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

Define

$$\mathbf{h}^i(\mathbf{W}) : \mathbb{R}^{r \times d} \rightarrow \mathbb{R}^d = \mathbf{E}_{\mathbf{x}}[f^i(\mathbf{W}\mathbf{x})\mathbf{x}]$$

$$\mathbf{M} := \mathbf{W}^{\top} \mathbf{W} \left( \frac{1}{k} \sum_{i=1}^k \mathbf{h}^i(\mathbf{W})(\mathbf{h}^i(\mathbf{W}))^{\top} \right) \mathbf{W}^{\top} \mathbf{W}.$$

## Representation learning - Result

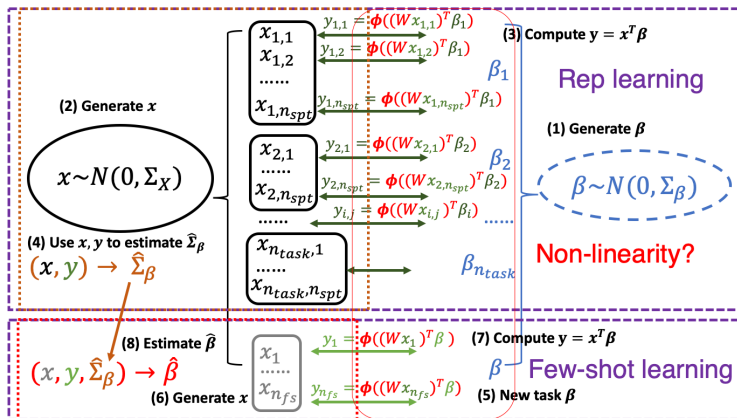
Suppose  $y\mathbf{x}$  is a subGaussian random vector with covariance upper bounded by  $\|\text{Cov}(y\mathbf{x})\| \leq \sigma^2$ . (These conditions hold when  $|f^i(\mathbf{x})| < \sigma$ .) Let  $\epsilon \in (0, 1)$ . Then there exists a constant  $c > 0$  such that

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

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

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

# Meta-learning - Nonlinear



## Few-shot learning - Metric

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

$$\mathcal{L}(f; \mathbf{P}) = \mathbf{E}_{\mathcal{P}_{\mathbf{x},y}} \text{loss}(f(\mathbf{P}\mathbf{x}), y)$$

$$\mathcal{L}_e(f; \mathbf{P}) = \frac{1}{n} \sum_{i=1}^n \text{loss}(f(\mathbf{P}\mathbf{x}_i), y_i).$$

We make the following assumption on the population risk.

1.  $\mathcal{L}$  is  $L$  Lipschitz in  $\mathbf{P}\mathbf{x}$ .
2.  $\min_{\mathbf{P}} \mathcal{L}(f; \mathbf{P}) = \mathcal{L}(f; \mathbf{W})$ .

## Few-shot learning - Metric

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

$$\mathcal{L}(f; \mathbf{P}) = \mathbf{E}_{\mathcal{P}_{\mathbf{x},y}} \text{loss}(f(\mathbf{P}\mathbf{x}), y)$$

$$\mathcal{L}_e(f; \mathbf{P}) = \frac{1}{n} \sum_{i=1}^n \text{loss}(f(\mathbf{P}\mathbf{x}_i), y_i).$$

We make the following assumption on the population risk.

1.  $\mathcal{L}$  is  $L$  Lipschitz in  $\mathbf{P}\mathbf{x}$ .
2.  $\min_{\mathbf{P}} \mathcal{L}(f; \mathbf{P}) = \mathcal{L}(f; \mathbf{W})$ .

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

$$\mathcal{L}(f; \mathbf{P}) = -\mathbf{E}_{\mathcal{P}_{\mathbf{x},y}} (y \log f(\mathbf{P}\mathbf{x}) + (1 - y) \log(1 - f(\mathbf{P}\mathbf{x})))$$

$$\mathcal{L}_e(f; \mathbf{P}) = -\frac{1}{n} \sum_{i=1}^n (y_i \log(f(\mathbf{P}\mathbf{x}_i)) + (1 - y_i) \log(1 - f(\mathbf{P}\mathbf{x}_i))).$$

## Few-shot learning

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

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



## Few-shot learning - Result

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

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

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

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

---

<sup>1</sup>bounded norm for Rademacher complexity analysis.