IDALAB

INTELLIGENT DATA ANALYTICS SALZBURG

PARIS
LODRON
UNIVERSITÄT
SALZBURG

Simon Hirländer

Tutorial RL4AA

Meta RL via gradients

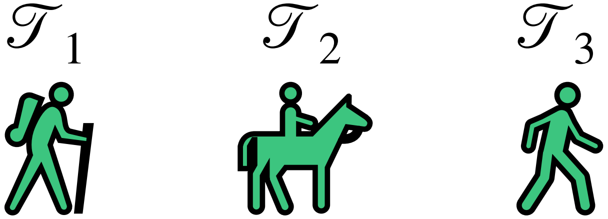
MAML outline

30

Require $p(\mathcal{T})$: distribution over tasks

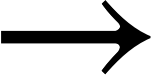
Require α, β: step size hyper-parameters

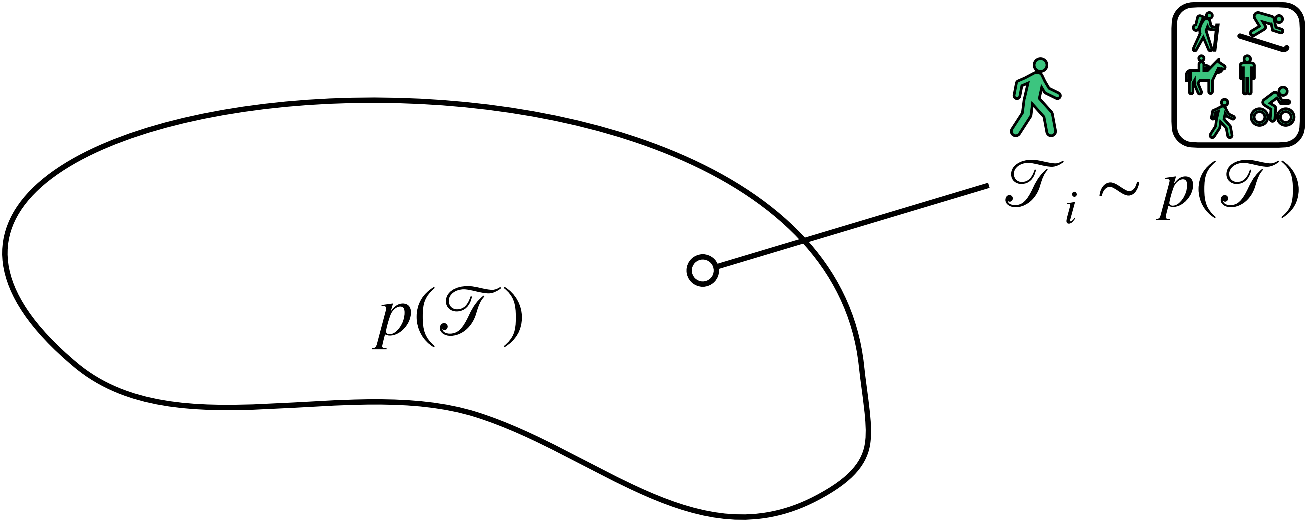
1. randomly initialise θ
2. **while** not done **do**
3.   sample batch of tasks $\mathcal{T}_i \sim p(\mathcal{T})$
4.   **for each** $\mathcal{T}_i$ **do**
5.   Sample $\mathcal{D}^{tr}_{\mathcal{T}_i} \sim \mathcal{D}_{\mathcal{T}_i}$
6.   Sample $\mathcal{D}^{test}_{\mathcal{T}_i} \sim \mathcal{D}_{\mathcal{T}_i}$
7.   Evaluate $\nabla_\theta \mathcal{L}(\theta, \mathcal{D}^{tr}_{\mathcal{T}_i})$ with respect to K examples
8.   Compute adapted parameters with gradient descent: $\phi_i = \theta - \alpha \nabla_\theta \mathcal{L}(\theta, \mathcal{D}^{tr}_{\mathcal{T}_i})$
9.   Update $\theta \leftarrow \theta - \beta \nabla_\theta \sum_i \mathcal{L}(\phi_i, \mathcal{D}^{test}_{\mathcal{T}_i})$

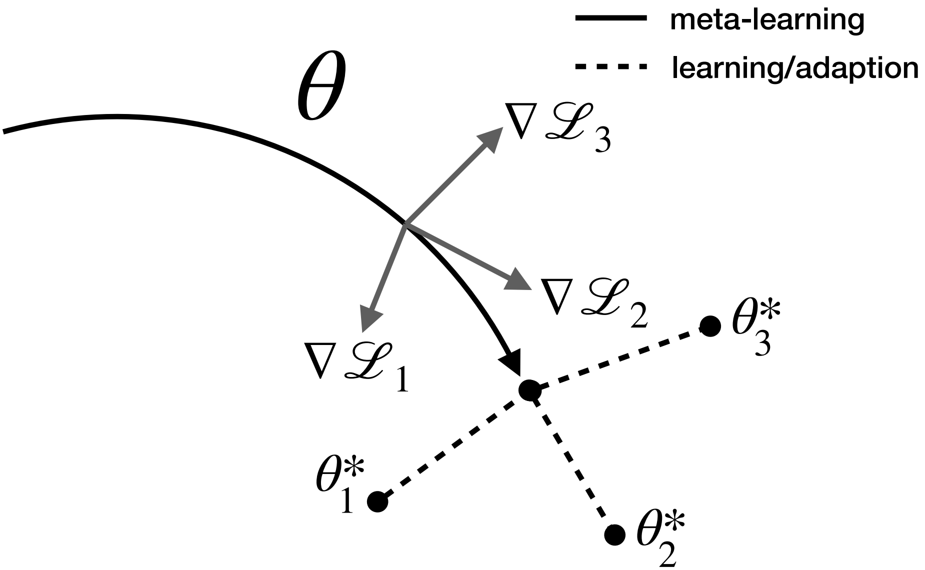


$\mathcal{T}_1$
$\mathcal{T}_2$
$\mathcal{T}_3$

$\mathcal{T}_{test}$




$p(\mathcal{T})$
$\mathcal{T}_i \sim p(\mathcal{T})$

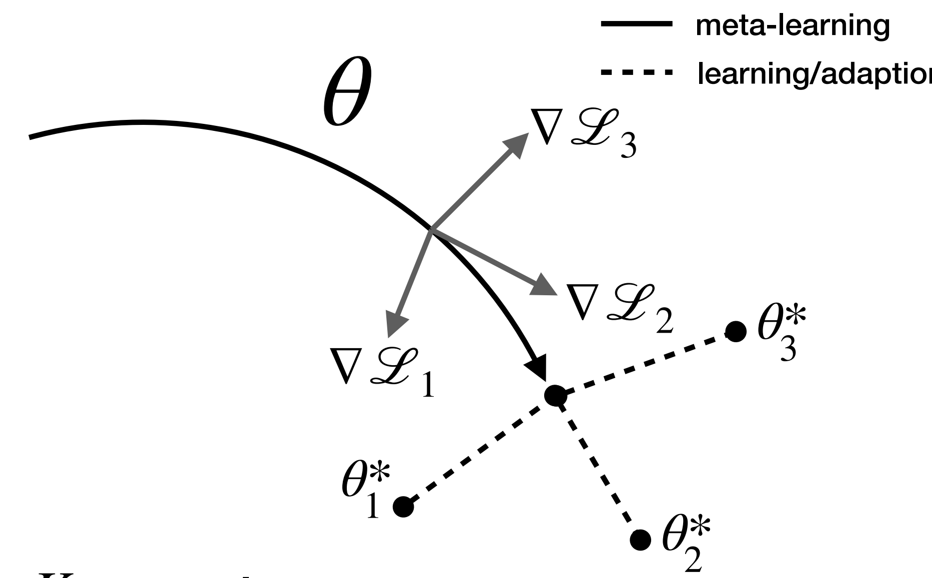

meta-learning
learning/adaption
θ
$\nabla \mathcal{L}_3$
$\nabla \mathcal{L}_2$
$\nabla \mathcal{L}_1$
θ_3^*
θ_1^*
θ_2^*

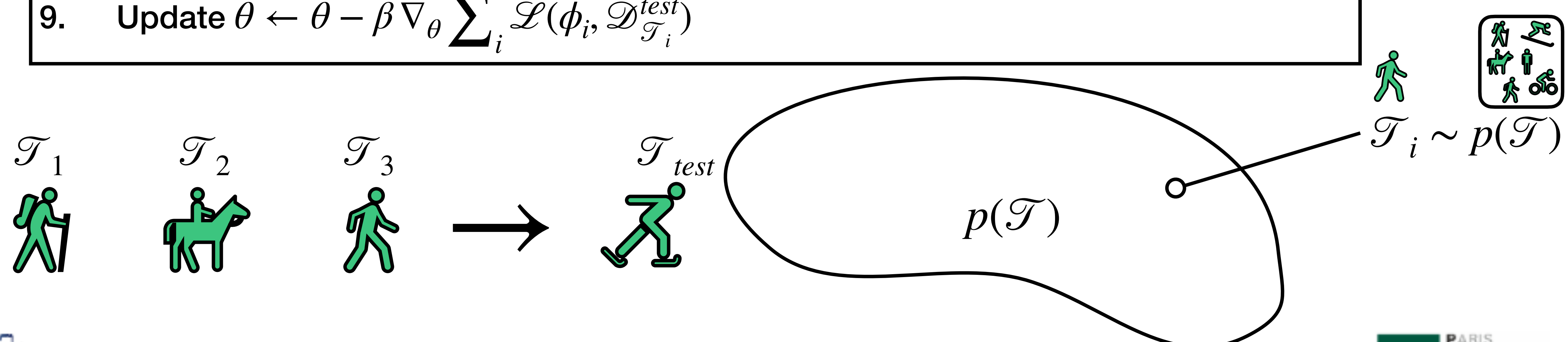
Meta RL via gradients

MAML outline

Require $p(\mathcal{T})$: distribution over tasks
Require α, β: step size hyper-parameters

1. randomly initialise θ
2. **while not done do**
3. sample batch of tasks $\mathcal{T}_i \sim p(\mathcal{T})$
4. **for each** $\mathcal{T}_i$ **do**
5. Sample $\mathcal{D}^{tr}_{\mathcal{T}_i} \sim \mathcal{D}_{\mathcal{T}_i}$
6. Sample $\mathcal{D}^{test}_{\mathcal{T}_i} \sim \mathcal{D}_{\mathcal{T}_i}$
7. Evaluate $\nabla_\theta \mathcal{L}(\theta, \mathcal{D}^{tr}_{\mathcal{T}_i})$ with respect to K examples
8. Compute adapted parameters with gradient descent: $\phi_i = \theta - \alpha \nabla_\theta \mathcal{L}(\theta, \mathcal{D}^{tr}_{\mathcal{T}_i})$
9. Update $\theta \leftarrow \theta - \beta \nabla_\theta \sum_i \mathcal{L}(\phi_i, \mathcal{D}^{test}_{\mathcal{T}_i})$





Why MAML is a good idea

- MAML is universally applicable beyond our specific scenario:
  - ➡ It can be implemented across various optimization problems.
  - ➡ The required gradients (to second order) can be efficiently computed using automatic differentiation.