Part 1

A Perspective View and Survey of Meta-Learning

By Ricardo Vilalta and Youssef Drissi
Presented by Rasheed Ali R.

What is Learning

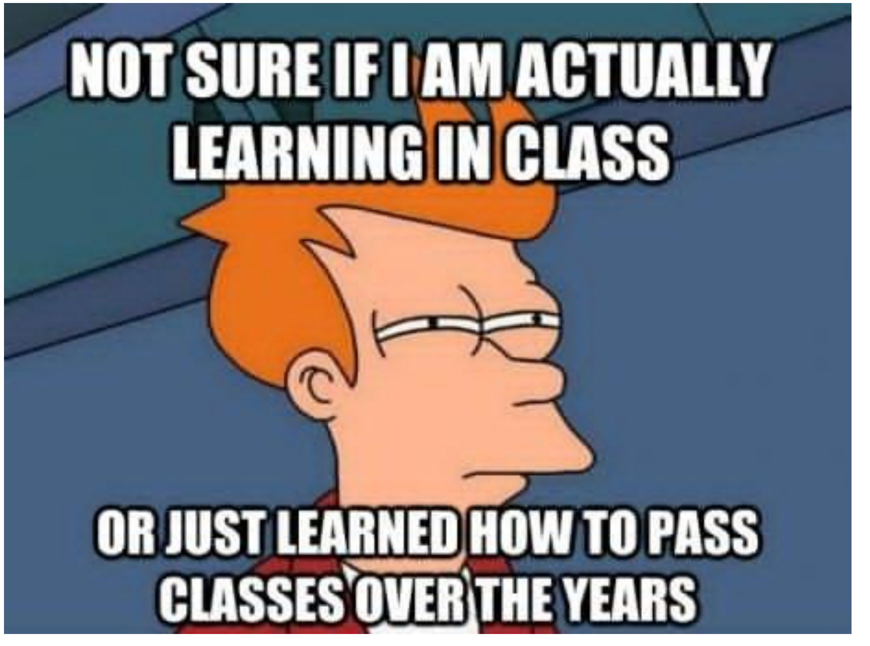
An agent is said to be learning if its performance improves with experience.

Limits of Learning

- Provided the same experience an agent will learn the same hypotheses
- An agent is unable to use experience from cross domains

What is Meta-Learning

"Meta-learning studies how to choose the right bias dynamically, as opposed to base-learning where the bias is fixed a priori, or user parameterized."



Structure

- Features: $\vec{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \in \mathcal{X}$
- Classes: $c \in \mathcal{C}$
- Target function: $f: \mathcal{X} \to \mathcal{C}$

Problem

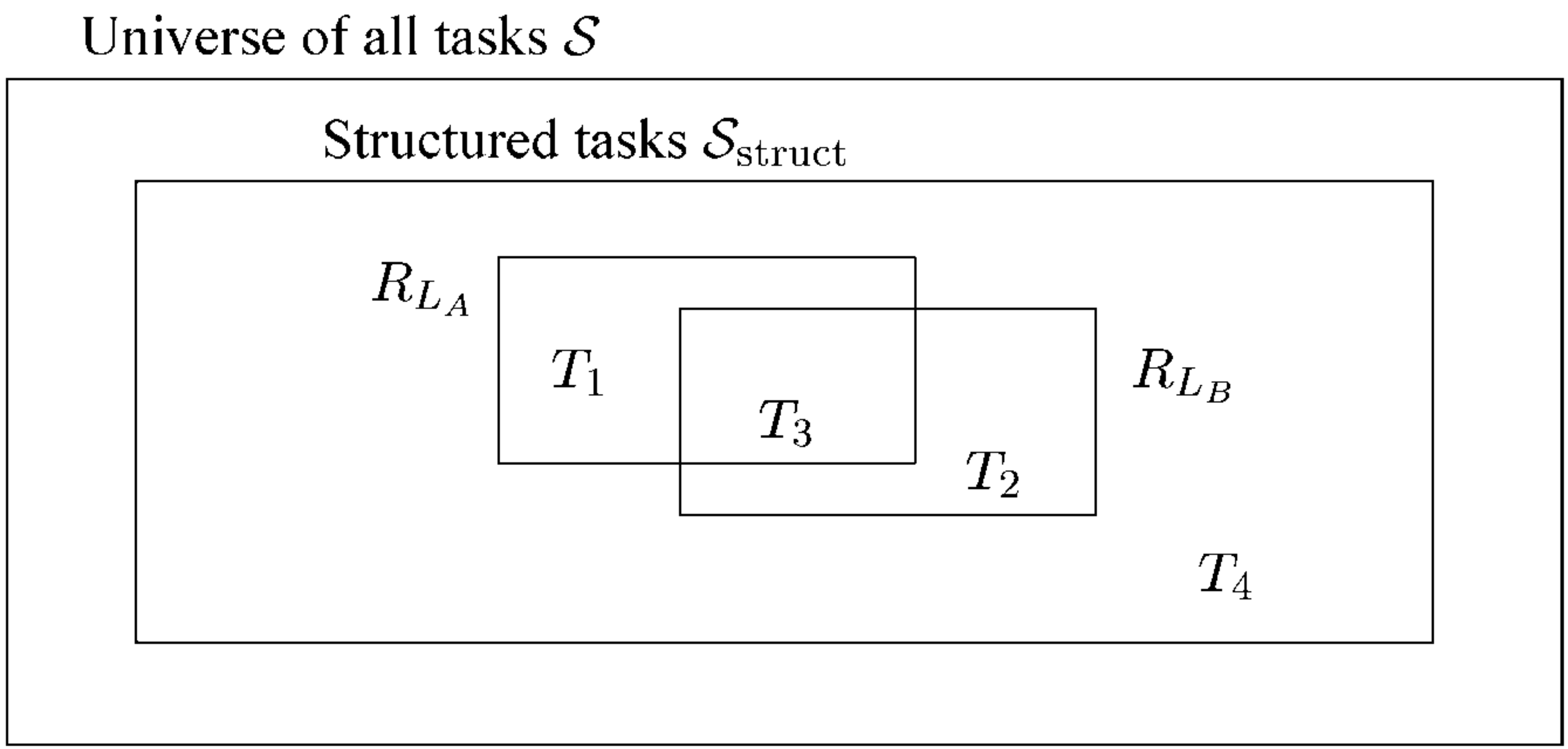
- Training Space: $\mathcal{T}_m \subset \mathcal{P}(\mathcal{X} \times \mathcal{C})$
- Training Set: $T_{train} = \{(\vec{x}_i, c_i)\}_{i=1}^{m} \in \mathcal{T}_m$
- Learning algorithm: $L: \mathcal{T} \rightarrow \mathcal{H}_L$
- Hypothesis: $h \in \mathcal{H}_L$
- Objective: $h \cong f$

Concepts

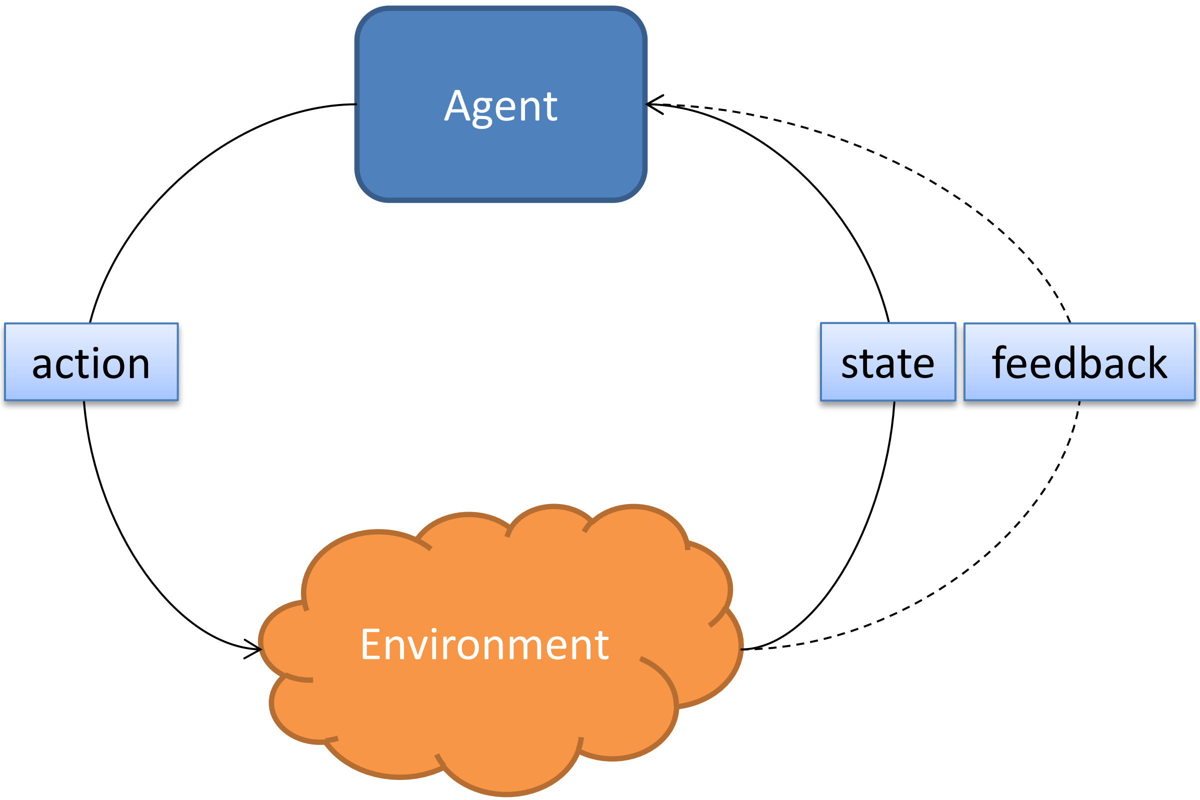
- Bias: assumptions restricting solution space
  1. Restricts size of $\mathcal{H}_L$
  2. Imposes preferences in $\mathcal{H}_L$
- Correct bias: $f \in \mathcal{H}_L$
- Stronger bias: $\left|\mathcal{H}_{L_A}\right| \leq \left|\mathcal{H}_{L_B}\right|$

Meta-Learning Goals

- In base-learning $\mathcal{H}_L$ is fixed
- Meta-learning adapts $\mathcal{H}_L$ based on experience
  1. Determine task properties that make L suitable
  2. Determine properties of L contributing to its dominance in particular tasks


Universe of all tasks $\mathcal{S}$
Structured tasks $\mathcal{S}_{\text{struct}}$
R_{L_A}
T_1
T_3
T_2
R_{L_B}
T_4

RL Example



Example Improvements

- Tuning learning parameters
- Refining interpretation of states

$$Q(s_t, a_t) \leftarrow \underbrace{Q(s_t, a_t)}_{\text{old value}} + \underbrace{\alpha_t(s_t, a_t)}_{\text{learning rate}} \times \left[\overbrace{\underbrace{R_{t+1}}_{\text{reward}} + \underbrace{\gamma}_{\text{discount factor}} \underbrace{\max_{a_{t+1}} Q(s_{t+1}, a_{t+1})}_{\text{max future value}}}^{\text{learned value}} - \underbrace{Q(s_t, a_t)}_{\text{old value}} \right]$$

General Approaches

- Meta*-learners
- Theoretically unbiased learners
  - Digital evolution
  - Analogy learning

Stacked Generalization

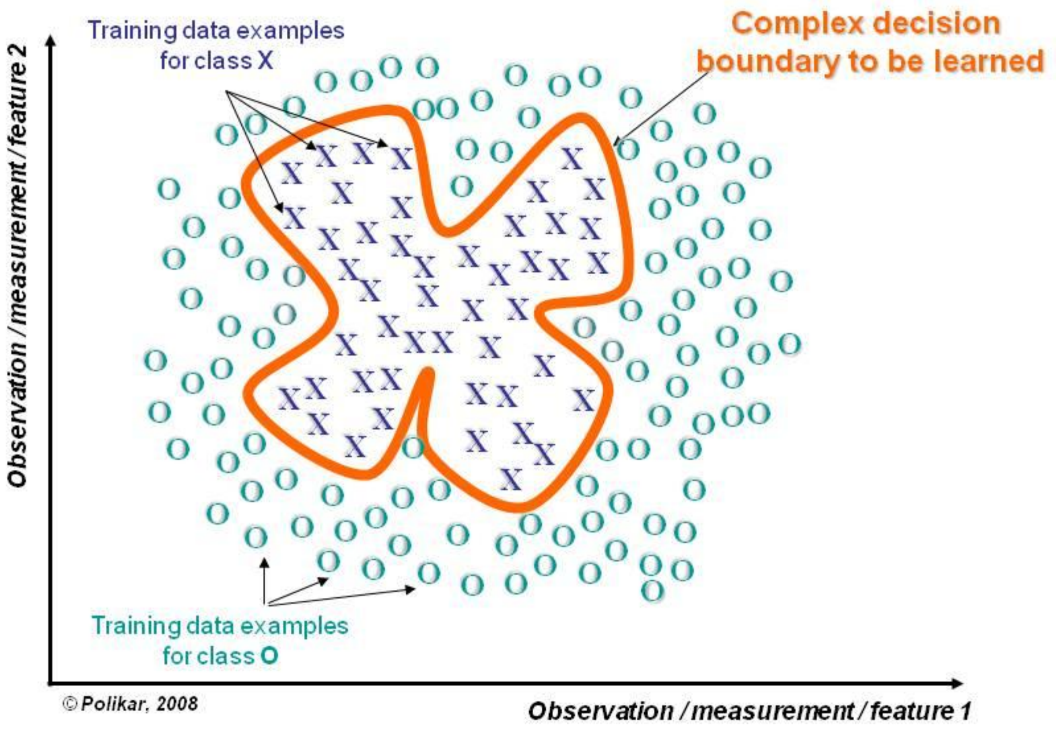
- $T_{train} = \{(\vec{x}_i, c_i)\}_{i=1}^{m}$
- Set of base-learners: $\{L_j\}_{j=1}^{q}$
- Meta-learner: Γ

Training Process

- $T_k = \{(\vec{x}_i, c_i)\}_{i \neq k}$
- $h_{jk} = L_j(T_k)$
- Meta-training set:

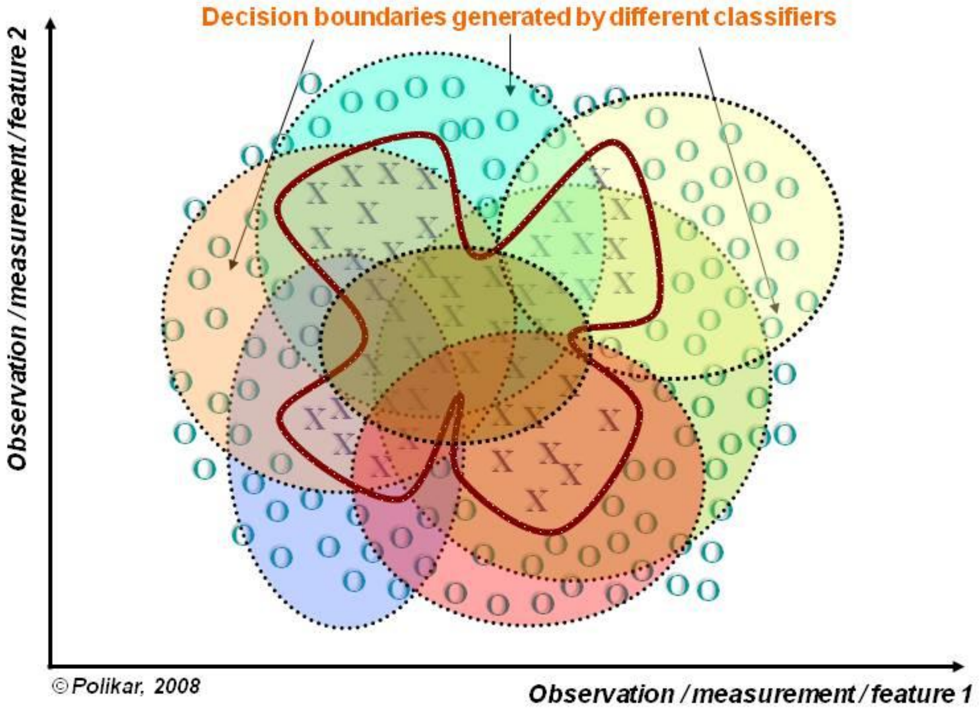
  $T' = \left\{\left(\left(h_{jk}(\vec{x}_k)\right)_{j=1}^{q}, c_k\right)\right\}_{k=1}^{m}$
- $h = \Gamma(T')$
- Classification: $h(\vec{x})$

Example



Training data examples
for class X
Complex decision
boundary to be learned
Training data examples
for class O
Observation / measurement / feature 2
Observation / measurement / feature 1
© Polikar, 2008

Example


Decision boundaries generated by different classifiers
Observation / measurement / feature 2
Observation / measurement / feature 1
© Polikar, 2008

Overview

- Learns from bias
- Predicts what is unpredicted by individual generalizers
- Generalizes winner takes all approaches
- Improves confidence
- Well suited for precision distributed and agent based systems

Conclusions

- Very effective
  - Outperforms Bayesian model-averaging
  - Outperforms majority voting approaches
- Used in difficult learning problems
  - Won $1 million Netflix learning competition