Lifelong Learning in Costly Feature Spaces

Maria-Florina Balcan, Avrim Blum, Vaishnavh Nagarajan
Lifelong Learning...

Building agents that learn like humans do...

Solve a series of related tasks efficiently by transferring knowledge through representations learned from previously-learned tasks.
Our goal: Feature-efficient (poly-time) lifelong learning algorithms for decision trees/lists, and real-valued polynomials with theoretical guarantees.
Related work

• **Knowledge transfer:**
  - Multi-task learning
  - Lifelong learning (mostly empirical)
    • Theoretical: Balcan et al. (2015), Pentina & Urner (2016)
    • Sample/computational efficiency

  Very little theoretical study of lifelong learning.

• **Budgeted learning**
  - predefined budget on feature evaluations
Outline

• Introduction
• Model
• Approach
• Main Results:
  • Decision trees
• More results:
  • Agnostic model
  • Lower bounds
Model

• Learn a **sequence** of $m$ (related) tasks/target functions, $g^{(i)}$ from data of $S$ samples each.
• Targets can be adversarially chosen.
• Each target maps from a common space of $N$ features.
• Focus in this talk:
  − Boolean decision trees of depth $d$
  − Each target = output of standard algorithm on dataset

\[
\begin{align*}
\text{(Unknown) Targets} & \quad g^{(1)} : \{0,1\}^N \rightarrow \{0,1\} \\
& \quad g^{(2)} : \{0,1\}^N \rightarrow \{0,1\} \\
& \quad \ldots \\
& \quad g^{(m)} : \{0,1\}^N \rightarrow \{0,1\}
\end{align*}
\]
Cost

• Total number of feature evaluations on training data across all \( m \) tasks

• Worst case cost: \( SmN \) by learning all targets “from scratch”:

No. of samples/task \((S)\) x No. of targets \((m)\) x No. of features \((N)\)

Data
\( S \times N \) matrices

\[ \begin{array}{c}
\text{Data} \\
S \times N \text{ matrices}
\end{array} = 1 \text{ unit cost} \]

Targets
\( (\text{Unknown}) \)

\( g^{(1)} : \{0,1\}^N \rightarrow \{0,1\} \)
\( g^{(2)} : \{0,1\}^N \rightarrow \{0,1\} \)
\( \ldots \)
\( g^{(m)} : \{0,1\}^N \rightarrow \{0,1\} \)

Time

Lifelong Learning in Costly Feature Spaces
A metafeature is a higher level concept i.e., higher level “building block” of a target function.

**Example:** A decision tree metafeature is a decision tree substructure without leaves.

Metafeature 1

Metafeature 2

concatenate & fill up leaves

and many more ...
**Target Relations**

A *metafeature* is a higher level concept i.e., higher level “building block” of a target function

**Example:** A decision tree metafeature is a decision tree substructure without leaves

Our belief is that the targets can be described using a common unknown set $\mathcal{F}$ of $K$ metafeatures. No. of metafeatures $(K) <<$ No. of features $(N)$ and no. of targets $(m)$
Outline

• Introduction
• Model
• Approach
• Main Results:
  • Decision trees
• More results:
  • Agnostic model
  • Lower bounds
Lifelong Learning Protocol

Targets

Learned representation = \( \tilde{\mathcal{F}} \)

\( g^{(3)}? \)

\{ Hypothesized metafeatures \}

Given subroutines \textbf{UseRep} and \textbf{ImproveRep}, for each task \( j \)

- Try \textbf{UseRep} i.e., use \( \tilde{\mathcal{F}} \) to evaluate very few features (<< \( N \)) per datapoint and learn a model that fits data.
Lifelong Learning Protocol

Given subroutines **UseRep** and **ImproveRep**, for each task $j$

- Try **UseRep** i.e., use $\tilde{F}$ to evaluate very few features ($<< N$) per datapoint and learn a model that fits data.
Lifelong Learning Protocol

Given subroutines **UseRep** and **ImproveRep**, for each task $j$:

- Try **UseRep** i.e., use $\tilde{F}$ to evaluate very few features ($<< N$) per datapoint and learn a model that fits data.
- If failed: learn from scratch (evaluate all $N$ features) and **ImproveRep** i.e., update $\tilde{F}$.
Lifelong Learning Protocol

Goal: Design **ImproveRep** and **UseRep** subroutines.

Learned representation = $\tilde{F}$

Hypothesized metafeatures

Targets
Outline

• Introduction
• Model
• Approach
• Main Results:
  • Decision trees
• More results:
  • Agnostic model
  • Lower bounds
Decision Trees: Result

Model: \( m \) tasks, \( N \) features, \( K \) metafeatures, \( d \) depth, \( S \) samples/task

Theorem (Decision trees): UseRep and ImproveRep together
1. learn at most \( K \) trees from scratch,
2. on the rest UseRep evaluates at most \( O(Kd) \) features per example \( \Rightarrow \) cost at most \( S \cdot O(KN + mKd) \)

Learning all targets from scratch costs \( S \cdot O(mN) \)
but recall:
no. of targets (\( m \)), no. of features (\( N \)) \( \gg \) no. of metafeatures (\( K \))
\( \Rightarrow mN \gg KN + mK = K(N + m) \)

Combinatorial challenge: Given many trees, find a small representation that describes them!
Decision Trees: **UseRep**

Model: \( m \) tasks, \( N \) features, \( K \) metafeatures, \( d \) depth, \( S \) samples/task

**UseRep Goal:** Learn a target \( g \) with few feature evaluations (\(< < N\)) per point if \( g \) can be described using \( \tilde{F} \)

\[
\tilde{F} = \{ x_3, x_3, x_7, x_5, x_{10}, \ldots \}
\]
**Decision Trees: UseRep**

Model: $m$ tasks, $N$ features, $K$ metafeatures, $d$ depth, $S$ samples/task

**UseRep Goal:** Learn a target $g$ with few feature evaluations ($<<N$) per point if $g$ can be described using $\tilde{F}$

**Key idea:** To determine feature with best split at a node, use $\tilde{F}$ to carefully select $|\tilde{F}|$ features to be evaluated on data.

$$\tilde{F} = \{ x_7, x_5, \ldots, x_{10} \}$$

Examine only: $\{ x_7, x_5 \}$
Decision Trees: ImproveRep

Model: \( m \) tasks, \( N \) features, \( K \) metafeatures, \( d \) depth, \( S \) samples/task

A. UseRep evaluates \( O(|\mathcal{F}| + d) \) features per example.

**ImproveRep Goal:** When UseRep fails, extract useful metafeature(s) from target learned from scratch.

**Key Idea:** Pick a path UseRep couldn’t learn.

Partial tree learned from UseRep

Correct tree from scratch
Decision Trees: **ImproveRep**

Model: $m$ tasks, $N$ features, $K$ metafeatures, $d$ depth, $S$ samples/task

A. **UseRep** evaluates $O(|\tilde{F}| + d)$ features per example.

**ImproveRep Goal:** When **UseRep** fails, extract useful metafeature(s) from target learned from scratch.

**Key Idea:** Pick a path **UseRep** couldn’t learn.
Decision Trees: ImproveRep

Model: \( m \) tasks, \( N \) features, \( K \) metafeatures, \( d \) depth, \( S \) samples/task

A. UseRep evaluates \( O(\tilde{F} + d) \) features per example.

B. ImproveRep adds \( d \) metafeatures in each call.

**Theorem (Decision trees):** UseRep and ImproveRep together
1. learn at most \( K \) trees from scratch,
2. on the rest UseRep evaluates at most \( O(Kd) \) features per example \( \Rightarrow \) cost at most \( S \cdot O(KN + mKd) \)

**Proof Idea:**
- One of the \( d \) metafeatures “approximately” recovers a new metafeature from underlying representation.
- After \( K \) calls of ImproveRep, UseRep never fails.
- Learned representation \( \tilde{F} \) has \( O(Kd) \) metafeatures.
Decision Trees: **ImproveRep**

Model: $m$ tasks, $N$ features, $K$ metafeatures, $d$ depth, $S$ samples/task

A. **UseRep** evaluates $O(|\tilde{\mathcal{F}}| + d)$ features per example.

B. **ImproveRep** adds $d$ metafeatures in each call.

**Theorem (Decision trees):** **UseRep** and **ImproveRep** together
1. learn at most $K$ trees from scratch,
2. on the rest **UseRep** evaluates at most $O(Kd)$ features per example $\Rightarrow$ cost at most $S \cdot O(KN+mKd)$

**More results:**
- for decision lists $O(S \cdot (KN+m(K^2+d)))$
- and for real-valued monomials/polynomials $O(S \cdot (KN+mK))$
Outline

• Introduction
• Model
• Approach
• Main Results:
  • Decision trees
• More results:
  • Agnostic model
  • Lower bounds
More results

Agnostic model: Learner faces $m + r$ targets where only some $m$ of which are related through $K$ metafeatures.

We design three algorithms:

- Fewer targets from scratch; Larger $\tilde{F}$ $O(KN + m(K+r))$
- A better balance $O(\sqrt{rKNm+mK})$
- More targets from scratch; Smaller $\tilde{F}$ $O(rKN + mK)$

Lower bounds on feature evaluations: When no. of unrelated targets $r$ is

- sufficiently small: our algorithms optimal in terms of $N$, $m$ and $K$: $\Omega(KN + mK)$
- too large: lifelong learning is meaningless $\Omega(mN)$
Conclusion

New insights into the lifelong learning paradigm:
• We propose a new metric of efficiency for costly feature spaces.
• We address combinatorial challenges in designing poly-time algorithms for decision trees/lists, monomials/polynomials.

Open questions:
• How do we recover the true decision tree representation exactly? How hard is it?
• Tighten the gap between lower and upper bounds for intermediate values of $r$ (no. of bad targets).
Thank you!

Questions?