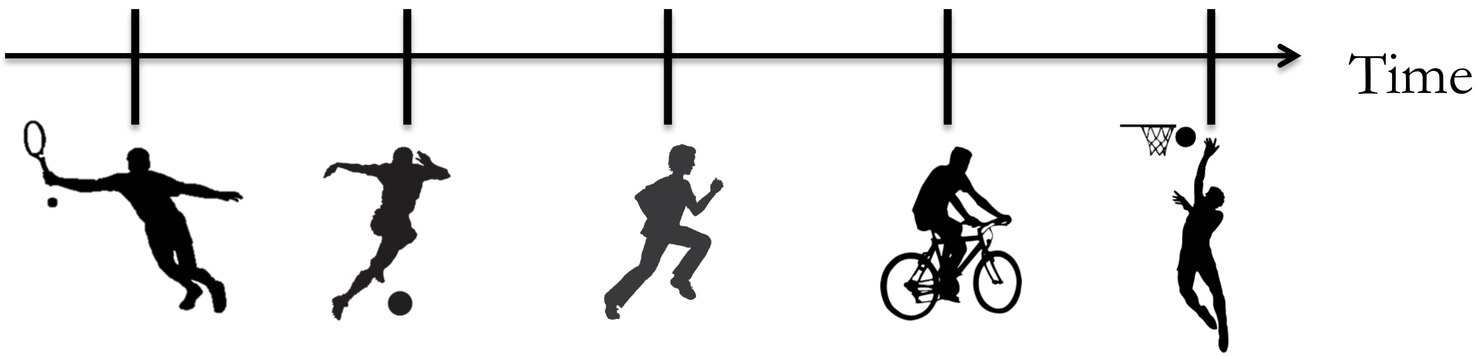
Lifelong Learning in Costly Feature Spaces

Maria-Florina Balcan, Avrim Blum,
Vaishnavh Nagarajan

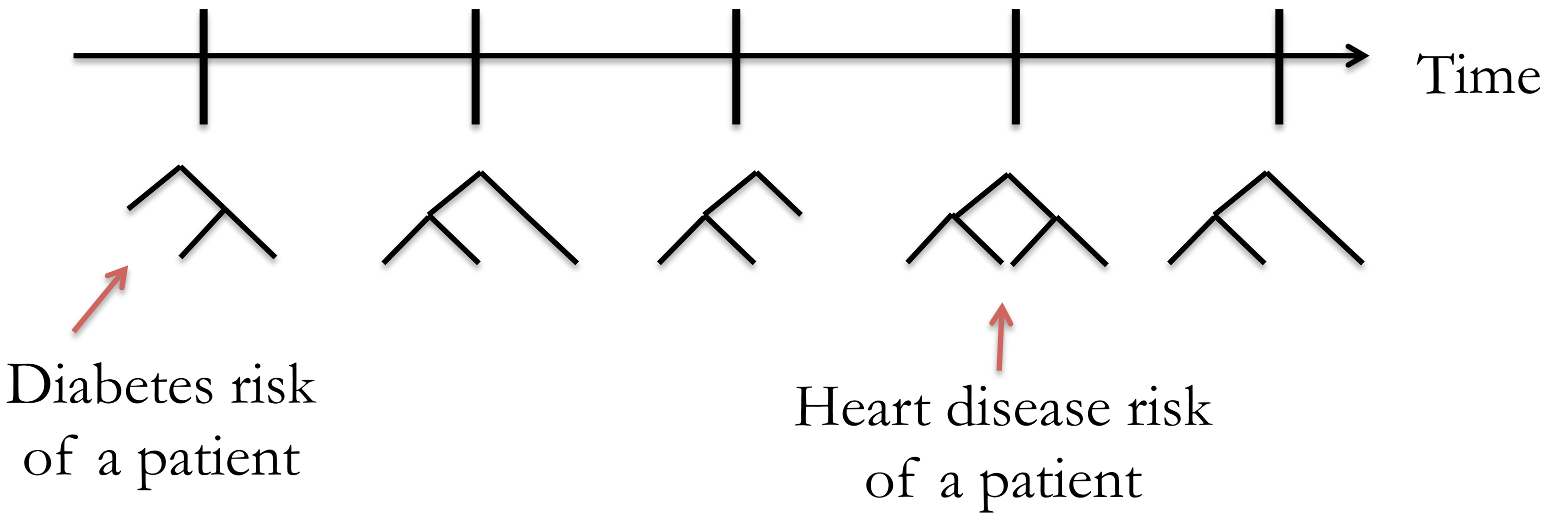
Lifelong Learning ...



Building agents that learn like humans do...

Solve a series of related tasks efficiently by transferring knowledge through representations learned from previously-learned tasks.

... in costly feature spaces



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.

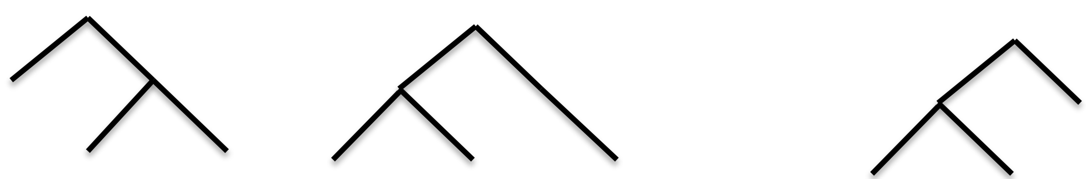
- Budgeted learning
  - predefined budget on feature evaluations

Outline

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

Model

- Learn a **sequence** of **m** (related) tasks/target functions, $\mathbf{g^{(j)}}$ 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



(Unknown) Targets $g^{(1)} : \{0,1\}^N \rightarrow \{0,1\}$ $g^{(2)} : \{0,1\}^N \rightarrow \{0,1\}$ … $g^{(m)} : \{0,1\}^N \rightarrow \{0,1\}$

Time

Cost

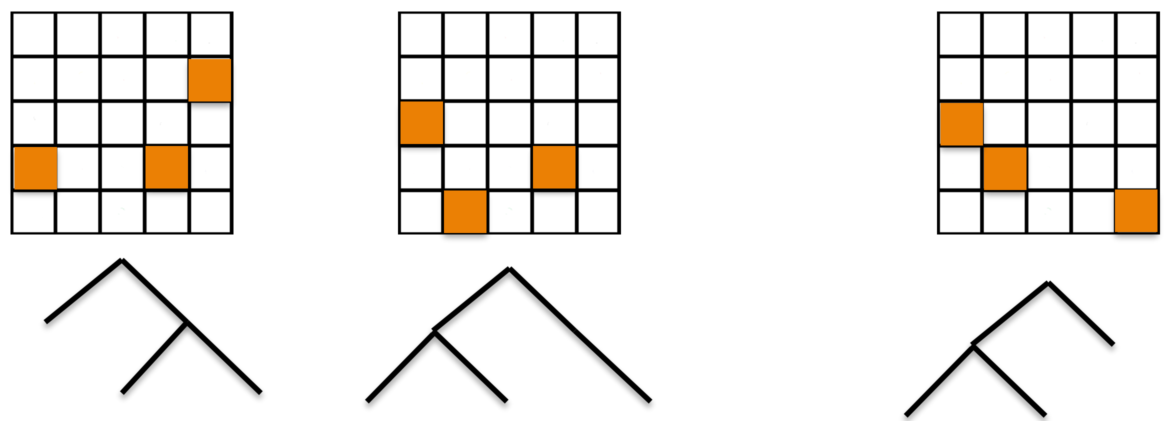
- Total number of feature evaluations on training data across all **m** tasks
- Worst case cost: **SmN** by learning all targets "from scratch":

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

Data

S x N matrices

■ = **1** unit cost



(Unknown) Targets

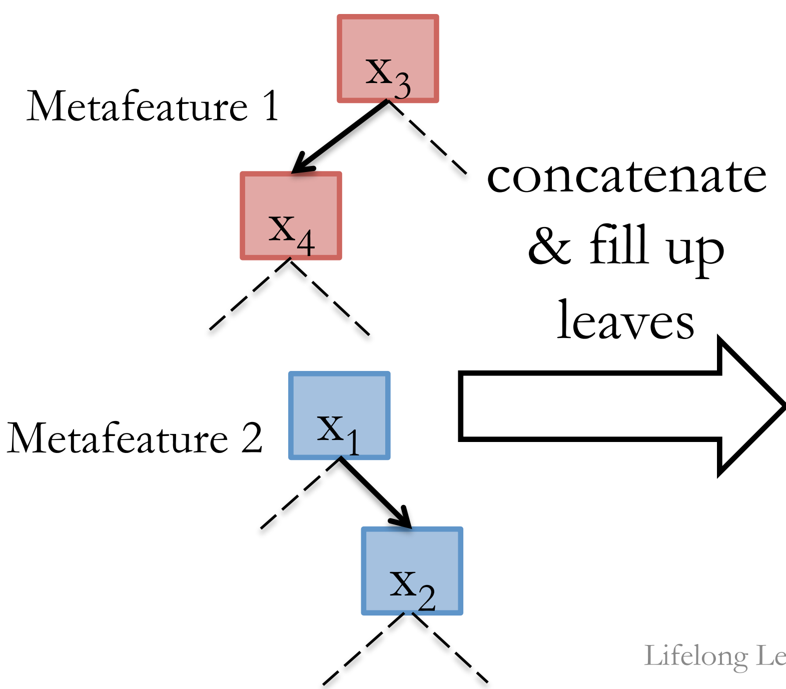
$g^{(1)} : \{0,1\}^N \rightarrow \{0,1\}$ $g^{(2)} : \{0,1\}^N \rightarrow \{0,1\}$ … $g^{(m)} : \{0,1\}^N \rightarrow \{0,1\}$

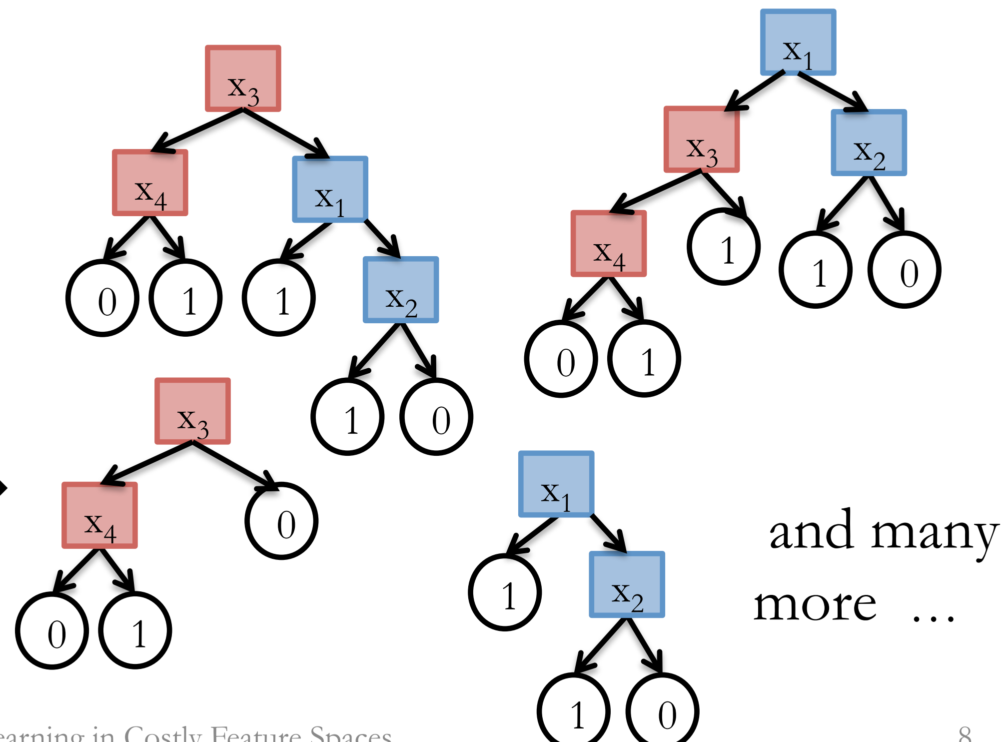
Time

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





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**)

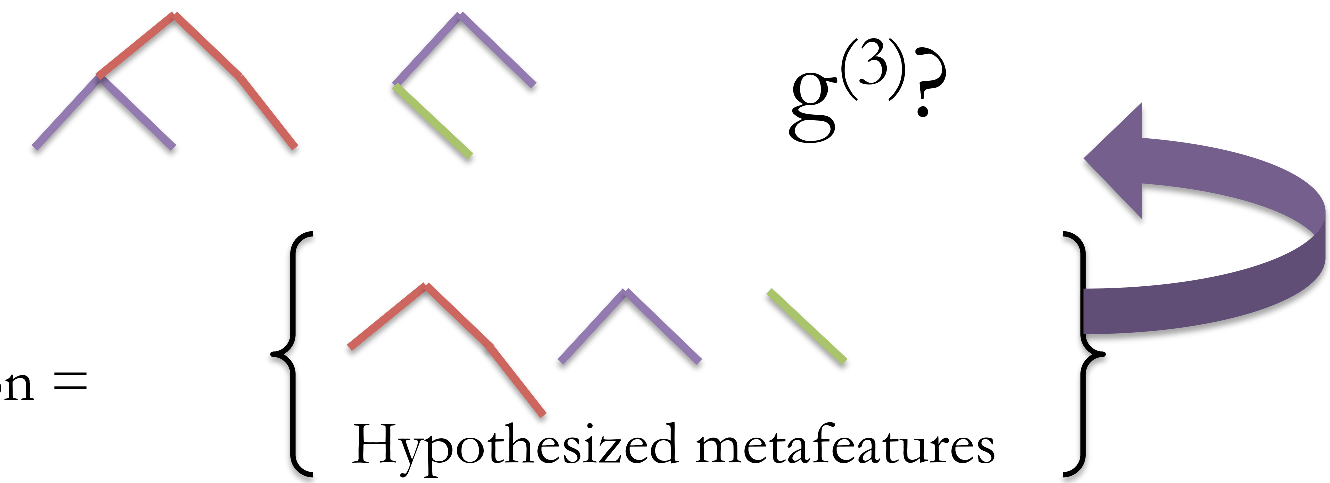
Outline

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

Lifelong Learning Protocol

Targets

Learned representation = $\tilde{\mathcal{F}}$



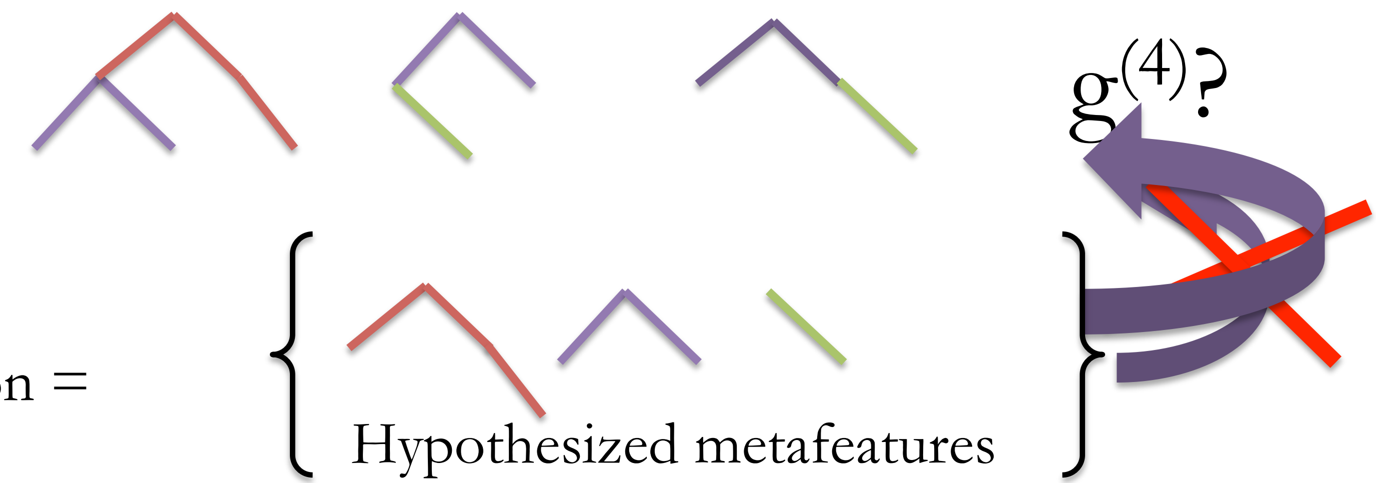
Given subroutines **UseRep** and **ImproveRep,** for each task j

- Try **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

Targets

Learned representation = $\tilde{\mathcal{F}}$



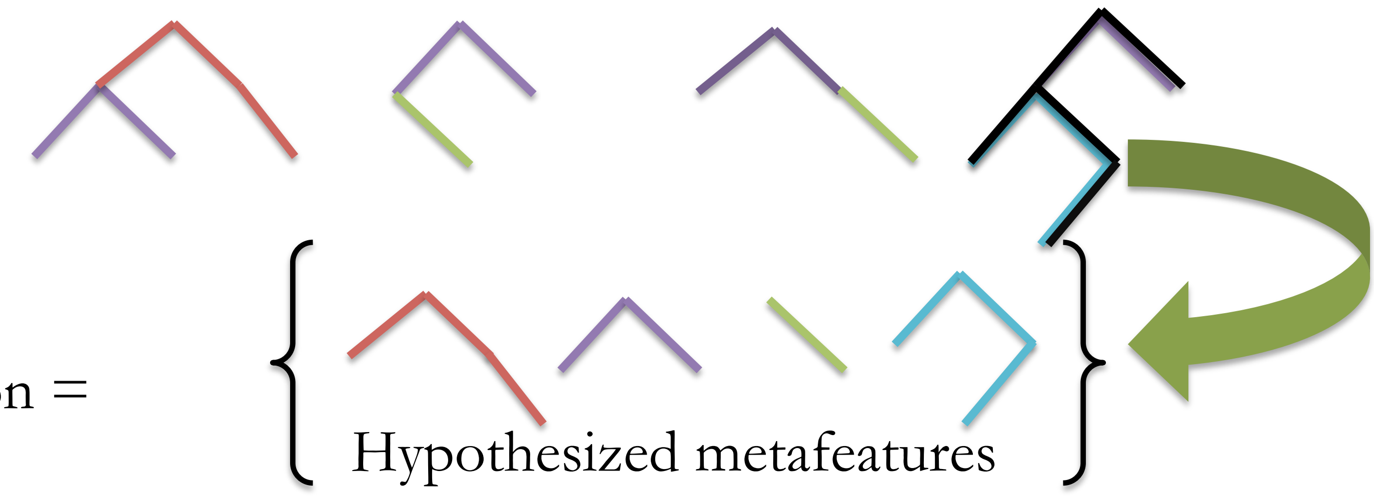
Given subroutines **UseRep** and **ImproveRep,** for each task j

- Try **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

Targets

Learned representation = $\tilde{\mathcal{F}}$



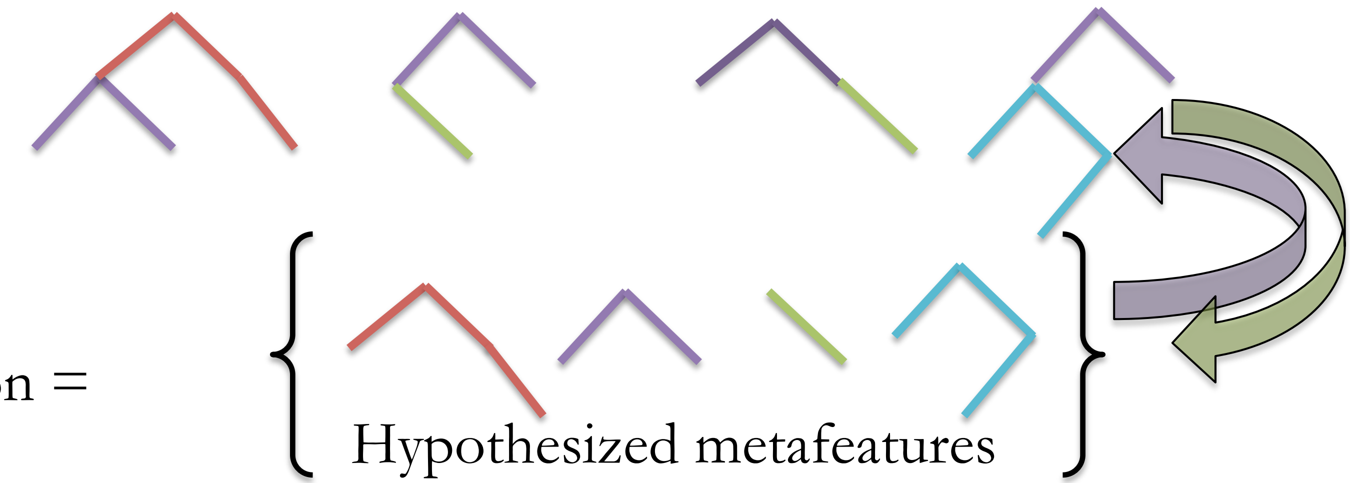
Given subroutines **UseRep** and **ImproveRep,** for each task j

- Try **UseRep** i.e., use $\tilde{\mathcal{F}}$ 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 $\tilde{\mathcal{F}}$

Lifelong Learning Protocol

Targets

Learned representation = $\tilde{\mathcal{F}}$



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

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 ➔ cost at most **S •O(KN+mKd)**

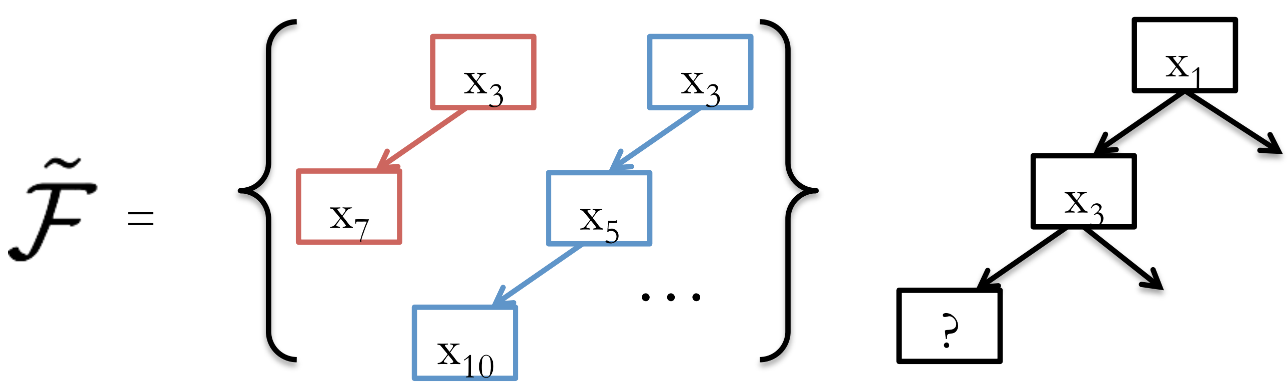
Learning all targets from scratch costs **S •O(mN)**
but recall:
no. of targets (**m**), no. of features (**N**) >> no. of metafeatures (**K**)
➔ **mN >> KN + mK = K(N+m)**

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 $\tilde{\mathcal{F}}$

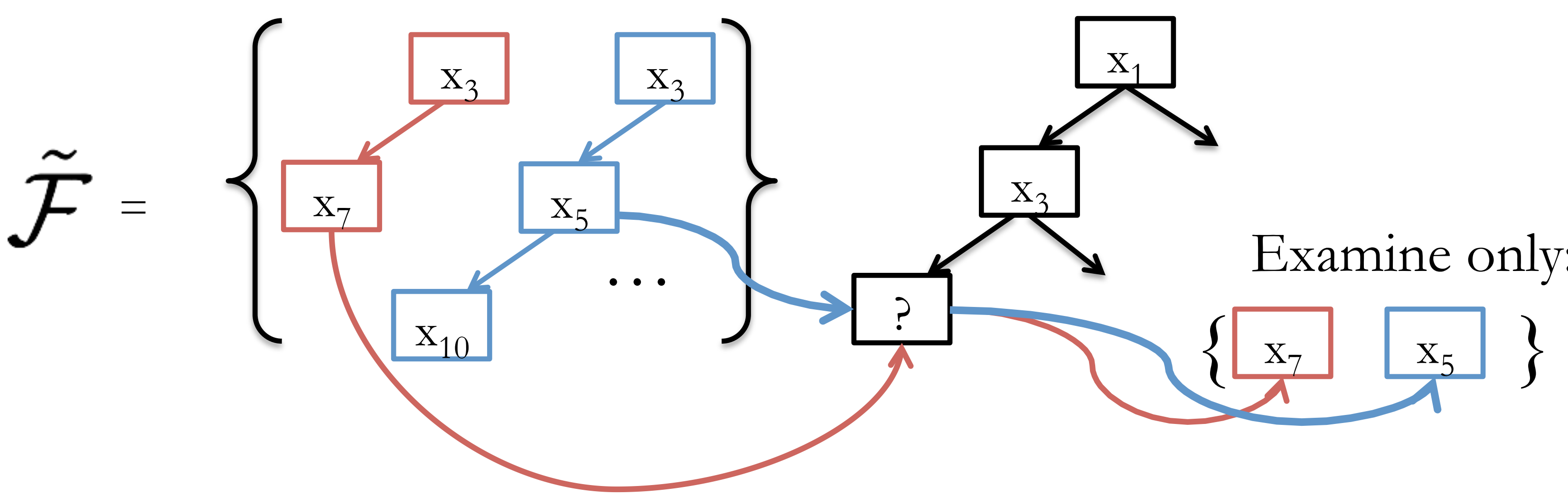


Decision Trees: UseRep

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 $\tilde{\mathcal{F}}$

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



Decision Trees: ImproveRep

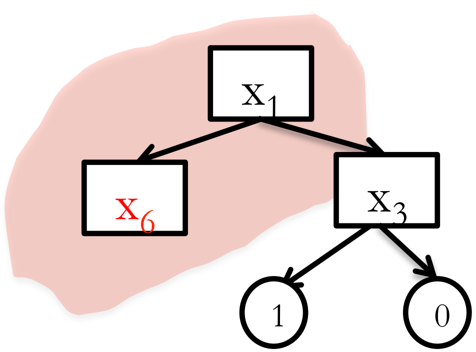
Model: **m** tasks, **N** features, **K** metafeatures, **d** depth, **S** samples/task

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

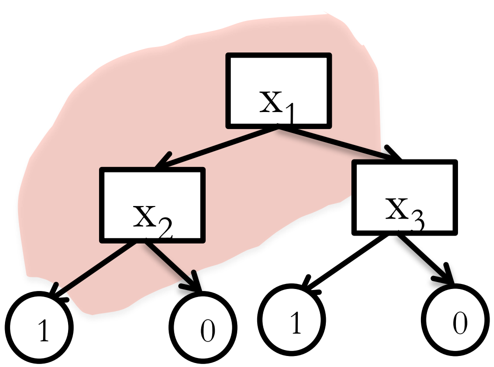
ImproveRep Goal: When UseRep fails, extract useful metafeature(s) from target learned from scratch.

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

Partial tree learned from **UseRep**



Correct tree from scratch



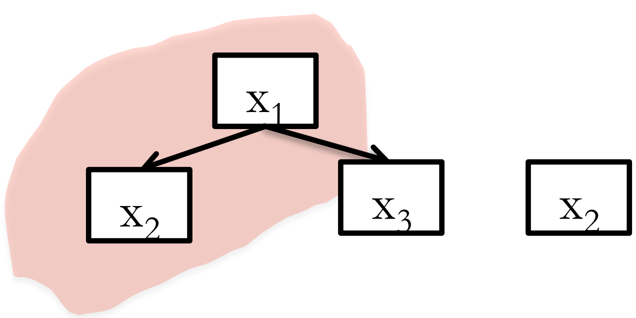
Decision Trees: ImproveRep

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

A. UseRep evaluates $\mathbf{O}(|\tilde{\mathcal{F}}| + \mathbf{d})$ 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(**$|\tilde{\mathcal{F}}|$**+ 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 ➔ cost at most **S •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 $\tilde{\mathcal{F}}$ 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 ➔ cost at most **S •O(KN+mKd)**

More results:

- for decision lists **O(S •(KN+m(**K^2**+d)))**
- and for real-valued monomials/polynomials **O(S •(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 $\tilde{\mathcal{F}}$ **O(KN+m(K+r))**

A better balance ⟷ **O($\sqrt{\text{rKNm}}$+mK)**

More targets from scratch; Smaller $\tilde{\mathcal{F}}$ **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?