AlphaD3M
Machine Learning Pipeline Synthesis

Iddo Drori, Yamuna Krishnamurthy, Remi Rampin, Raoni de Paula Lourenco, Jorge Piazentin Ono, Kyunghyun Cho, Claudio Silva, Juliana Freire

NEW YORK UNIVERSITY

International Workshop on Automatic Machine Learning, ICML, 2018

Automatic Machine Learning: Learning to Learn

Input: dataset, well defined task, performance criteria.

Goal: find best solution of task with respect to dataset.

Motivation: Dual Process Iteration and Self Play

Dual process theory: Thinking fast and slow, Daniel Kahneman (2002 Nobel Prize in Economics).

Expert iteration: Thinking fast and slow with deep learning and tree search, Anthony et al., NIPS 2017.

AlphaZero, self-play: Mastering chess and Shogi by self-play with a general reinforcement learning algorithm, Silver et al., NIPS 2017.

Single player AlphaZero with sequence model: AlphaD3M.

Single player AlphaZero, backwards: Solving the Rubik's cube without human knowledge, McAleer et al., 5.2018.

Min-max optimization, Nash equilibrium: Dual Policy Iteration, Sun et al., 5.2018.

Motivation: Dual Process Theory

Type 1

Autonomous

Does not require working memory

Type 2

Involves mental simulation and decoupling

Requires working memory

Dual Process Theory: Simple Analogy

Q: What is 34 squared?

Dual Process Theory: Simple Analogy

Type 1

30 x 30 = 900   4 x 30 = 120   30 x 4 = 120   4 x 4 = 16

Type 2

34 x 34 = 34 x 30 + 34 x 4   34 x 30 = 30 x 30 + 4 x 30   34 x 4 = 30 x 4 + 4 x 4

Dual Process Theory: Simple Analogy

A: 1156

Dual Process Theory: Simple Analogy

Q: Second time, what is 34 squared?

A: 1156 right away, since its now type 1, so we’ll keep the network which knows this rather than previous network.

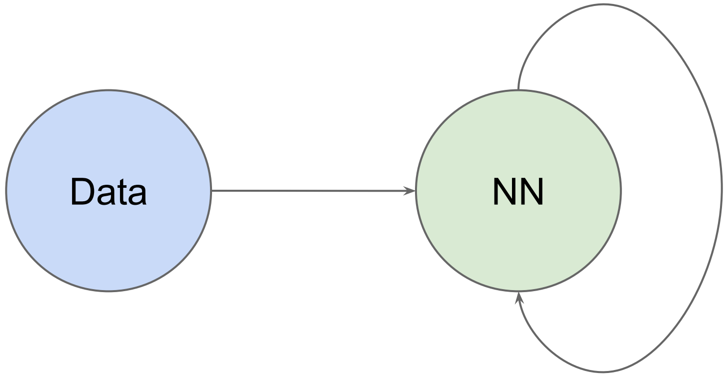
Q: Next, what is 34^4, use 34 squared etc.

Dual process iteration with self play.

Neural Network

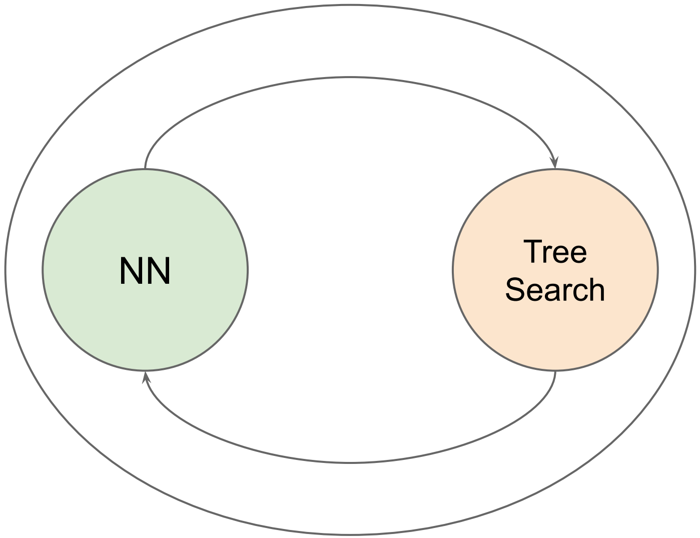
Stochastic Gradient Descent, forward and backward passes

Iterative type 1 architecture



Expert Iteration

Thinking fast and slow with deep learning and tree search, Anthony et al., NIPS 2017.



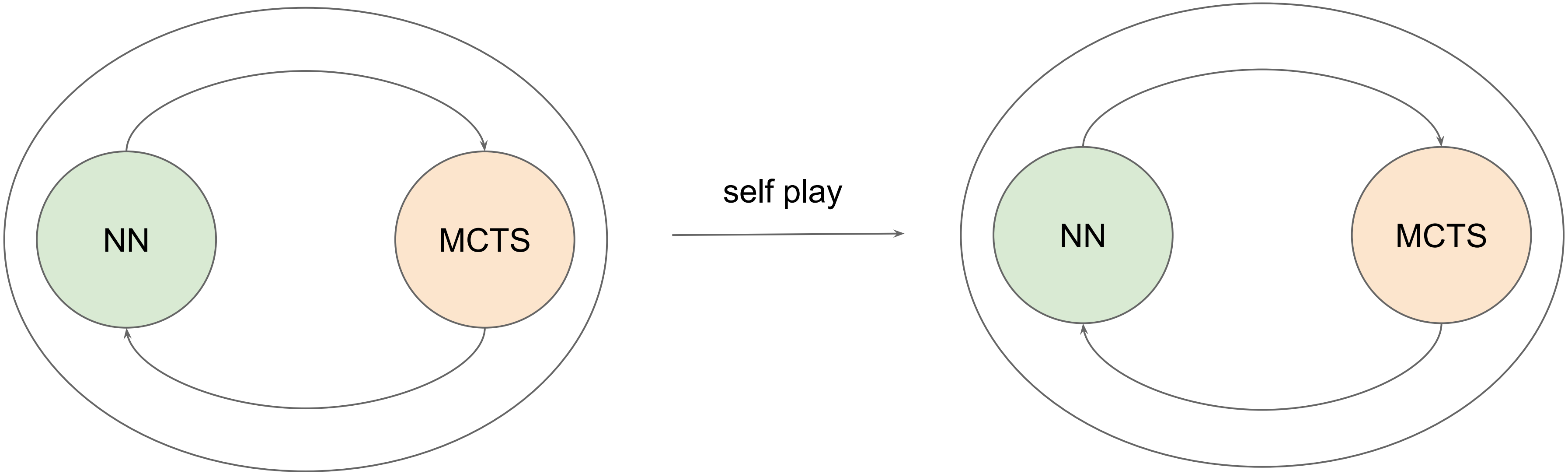
Type 2

Tree search cannot be **efficiently** replaced by type 1 NN's: Learning to search with MCTSnets (Guez et al, ICLR 2018).

Humans use NN's for type 2, slowly.

AlphaZero

Mastering chess and shogi by self-play with a general reinforcement learning algorithm, Silver et al., NIPS 2017.



AutoML Methods

Differentiable programming: End-to-end learning of machine learning pipelines with differentiable primitives (Milutinovic et al, AutoDiff 2017). Type 1 process only.

Bayesian optimization, hyperparameter tuning: Autosklearn (Feurer et al, NIPS 2015), AutoWEKA (Kotthoff et al, JMLR 2017),

Tree search of algorithms and hyperparameters, multi-armed bandit: Auto-Tuned Models (Swearingen et al, Big Data 2017)

Evolutionary algorithms: TPOT (Olson et al, ICML 2016) represent machine learning pipelines as trees, Autostacker (Chen et al, GECCO 2018) represent machine learning pipelines as stacked layers.

Data Driven Discovery of Models (D3M)

DARPA D3M project: infrastructure to automate model discovery.

Goal: solve any task on any dataset specified by a user.

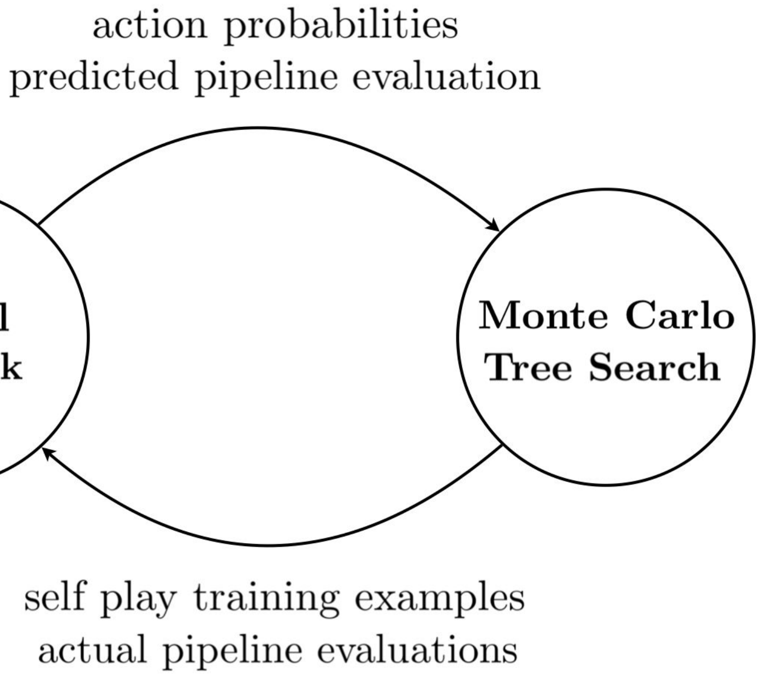
1. Broad set of computational primitives as building blocks.
2. Automatic systems for machine learning, synthesize pipeline and hyperparameters to solve a previously unknown data and problem.
3. Human in the loop: user interface that enables users to interact with and improve the automatically generated results.

Pipelines: pre-processing, feature extraction, feature selection, estimation, post-processing, evaluation.

AlphaD3M Single Player Game Representation

	AlphaZero	AlphaD3M
Game	Go, chess	AutoML
Unit	piece	pipeline primitive
State	configuration	meta data, task, pipeline
Action	move	insert, delete, replace
Reward	win, lose, draw	pipeline performance

AlphaD3M Iterative Improvement



Neural Network

Type 1: Optimize loss function by stochastic gradient descent.

Optimize network parameters θ: make predicted model S match real world model R, and predicted evaluation v match real evaluation e.

$$f_\theta(s) = (P(s,a), v(s))$$

$$L(\theta) = S \log R + (v - e)^2 + \alpha ||\theta||_2 + \beta ||S||_1$$

Monte Carlo Tree Search

Type 2 using Type 1: MCTS calling NN action value function

$$U(s,a) = Q(s,a) + cP(s,a)\frac{\sqrt{N(s)}}{1+N(s,a)}$$

Q(s,a): expected reward for action a from state s

N(s,a): number of times action a was taken from state s

N(s): number of times state s was visited

P(s,a): estimate of neural network for probability of taking action a from state s

c: constant determining amount of exploration

Pipeline Encoding

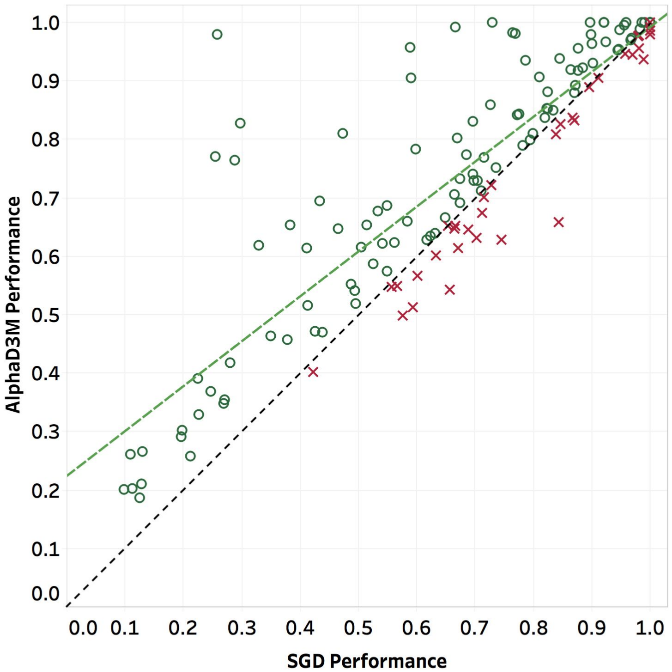
Our architecture **models meta data, task and entire pipeline chain as state** rather than individual primitives.

Given datasets D, tasks T, and a set of possible pipeline sequences $S_1, \ldots, S_n$, from the available machine learning, and data pre and post processing primitives.

- For each dataset D_i and task T_j:
  1. Encode dataset D_i as meta data features $f(D_i)$.
  2. Encode task T_j.
  3. Encode the current pipeline at time t by a vector S_t.
  4. Encode action $f_a(S_t)$, so policy π maps $(f(D_i), T_j, S_t)$ to $f_a(S_1), \ldots, f_a(S_n)$.

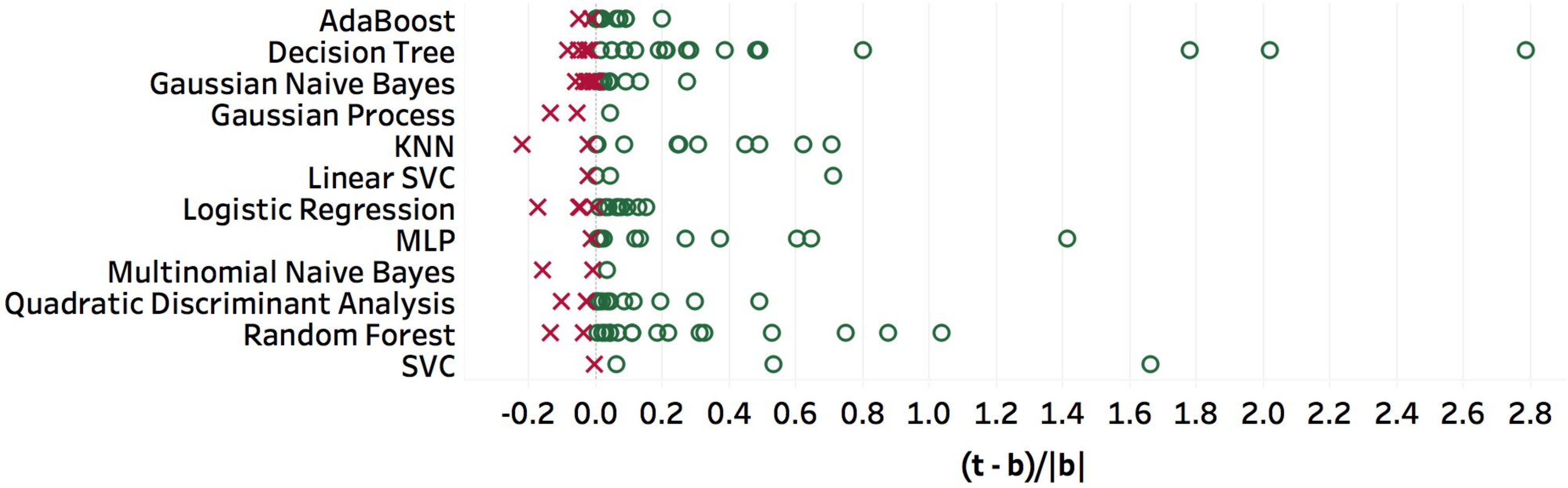
AlphaD3M vs. SGD Performance on OpenML

SGD baseline: classification with feature selection

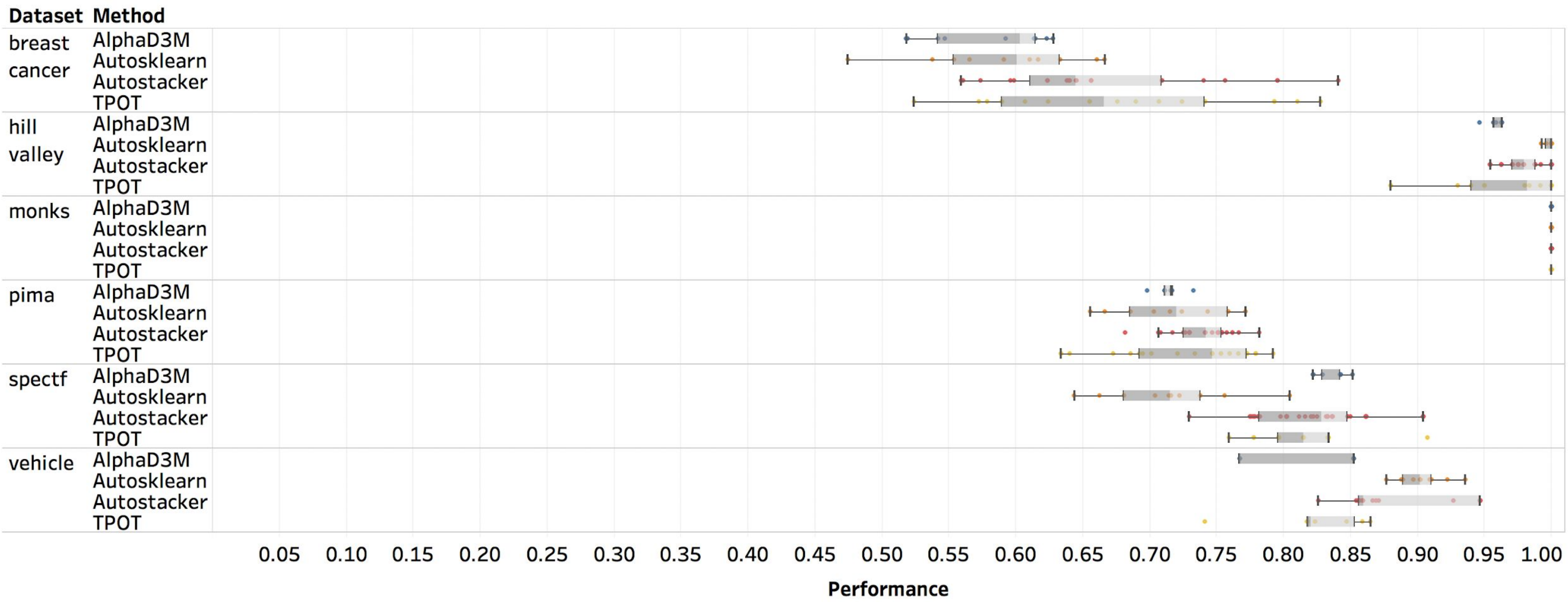


AlphaD3M vs. SGD for Different Estimators

Comparison of normalized AlphaD3M performance t and SGD baseline performance b, by estimator.



Comparison of AutoML Methods on OpenML



AlphaD3M Running Time Comparison

AlphaD3M implementation utilizes 4 Tesla P100 GPU's for NN.

Each experiment runs 10 times computing mean and variance.

Dataset/Method	TPOT	Autostacker	AlphaD3M	Speedup vs TPOT	Speedup vs AS
breast cancer	3366	1883	460	7.3	4
hill valley	17951	8411	556	32.2	15.1
monks	1517	1532	348	4.3	4.3
pima	5305	1940	619	8.5	3.1
spectf	4191	1673	522	8	3.2
vehicle	16795	4010	531	31.6	7.5

Conclusions

Automatic machine learning: competitive performance, order of magnitude faster than existing methods.

First single player AlphaZero game representation.

Meta learning by modeling meta-data, task, and entire pipelines as state.

Acknowledgements

This work has been supported in part by the Defense Advanced Research Projects Agency (DARPA) Data-Driven Discovery of Models (D3M) Program.

Thank you

Iddo Drori, Yamuna Krishnamurthy, Remi Rampin, Raoni de Paula Lourenco, Jorge Piazentin Ono,
Kyunghyun Cho, Claudio Silva, Juliana Freire

NEW YORK UNIVERSITY

International Workshop on Automatic Machine Learning, ICML, 2018