Algorithm Selection

Predict which algorithm to use!

AutoML.org

UNI FREIBURG

The Problem

Availability of Algorithms

Machine Learning

SVM

K-nearest Neighbor

Random Forest

Gradient Boosting

Deep Neural Network

Satisfiability Solving

lingeling

cryptominisat

glucose

probSAT

CaDiCaL

syrup

Sorting

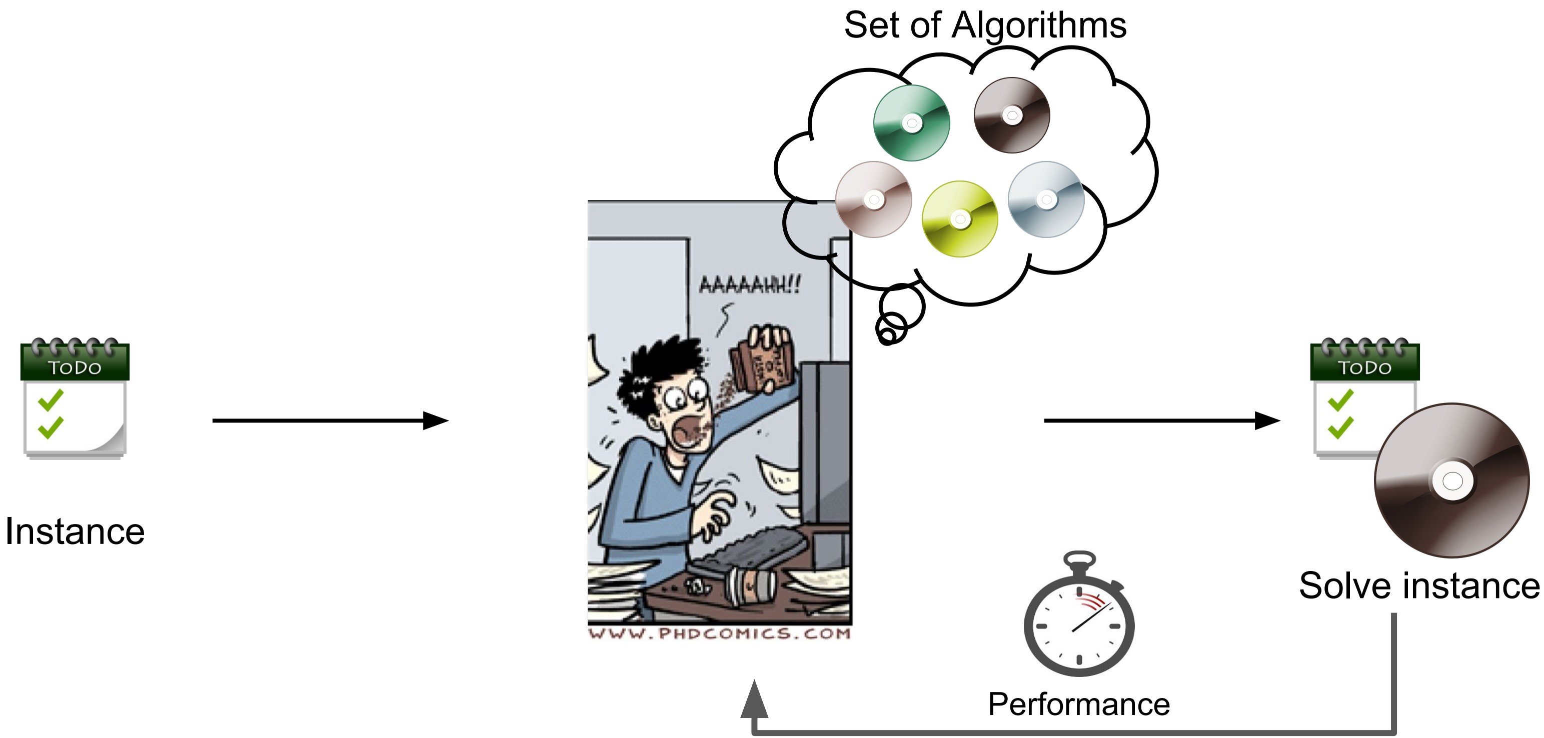
Merge Insertion

Quick sort

Merge sort

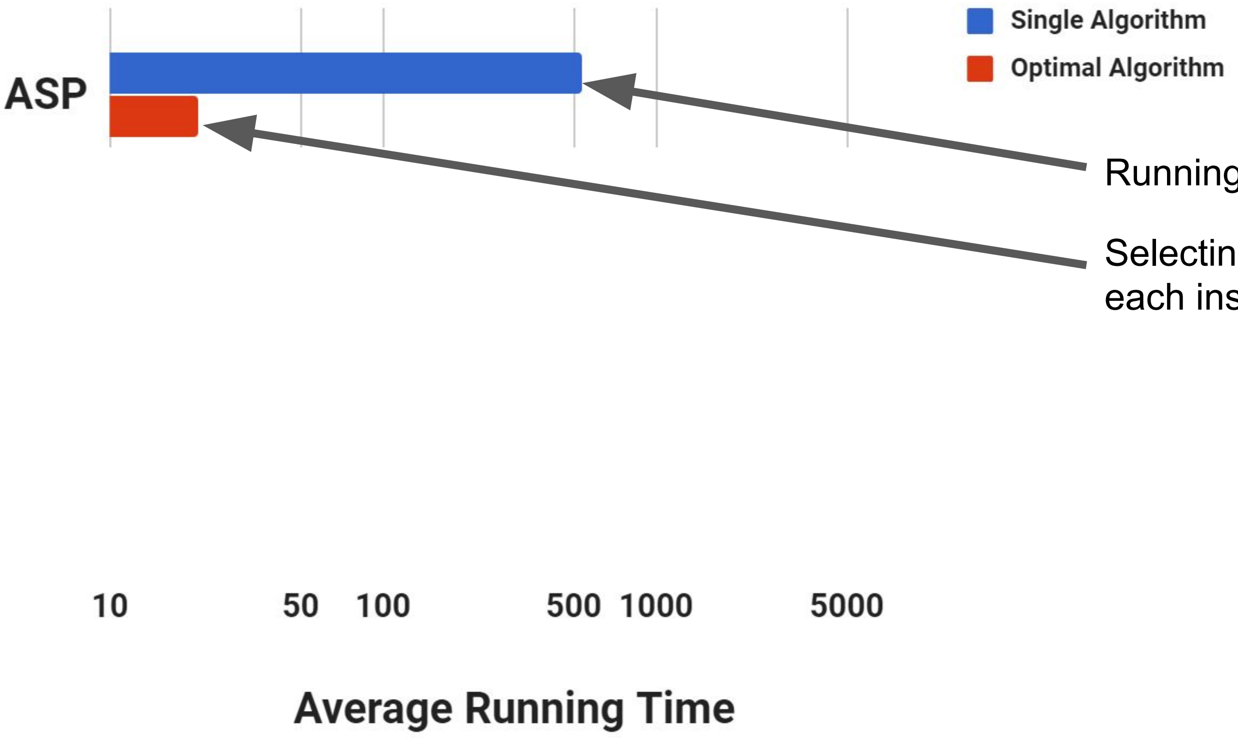
Binary tree sort

Manual Algorithm Selection



Goal: Select the algorithm with the best performance for a given instance

Algorithm Selection Matters!

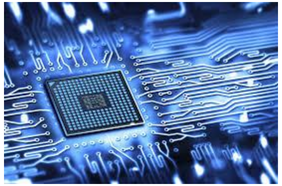


Running always the same algorithm

Selecting the best algorithm for each instance!

Open Algorithm Selection Challenge 2017

[**Lindauer** et al. AIJ 2019]

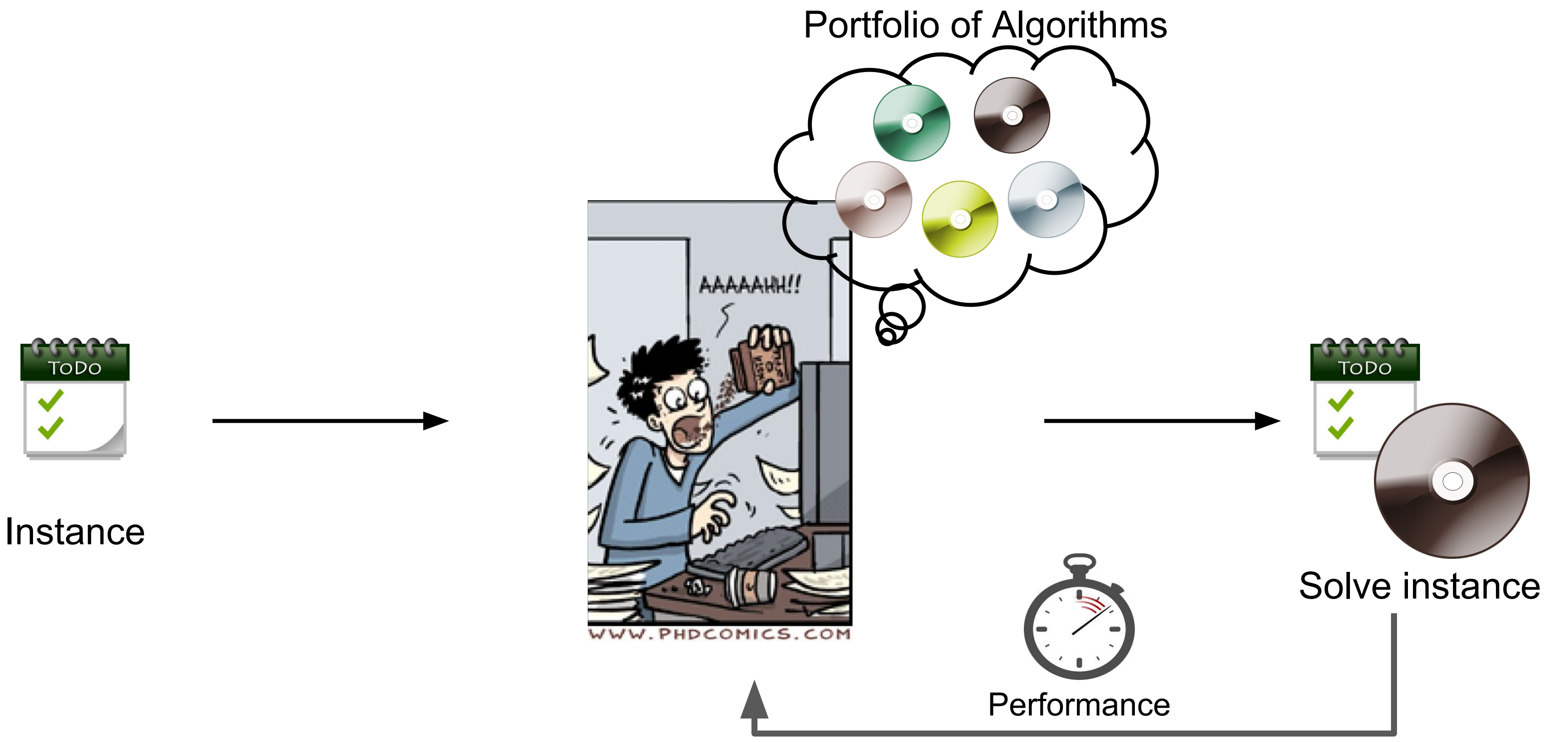


Domain	Average opt. Speedup
Mixed Integer Programming (MIP)	10
Maximum Satisfiability Problem (MAXSAT)	15
Boolean Satisfiability Problem (SAT)	30
Structure learning in Bayesian networks	41
Constraint Satisfaction Problem (CSP)	61
Quantified Boolean Formula (QBF)	**264**

Machine Learning (OPENML-Weka; absolute impr.)	2%

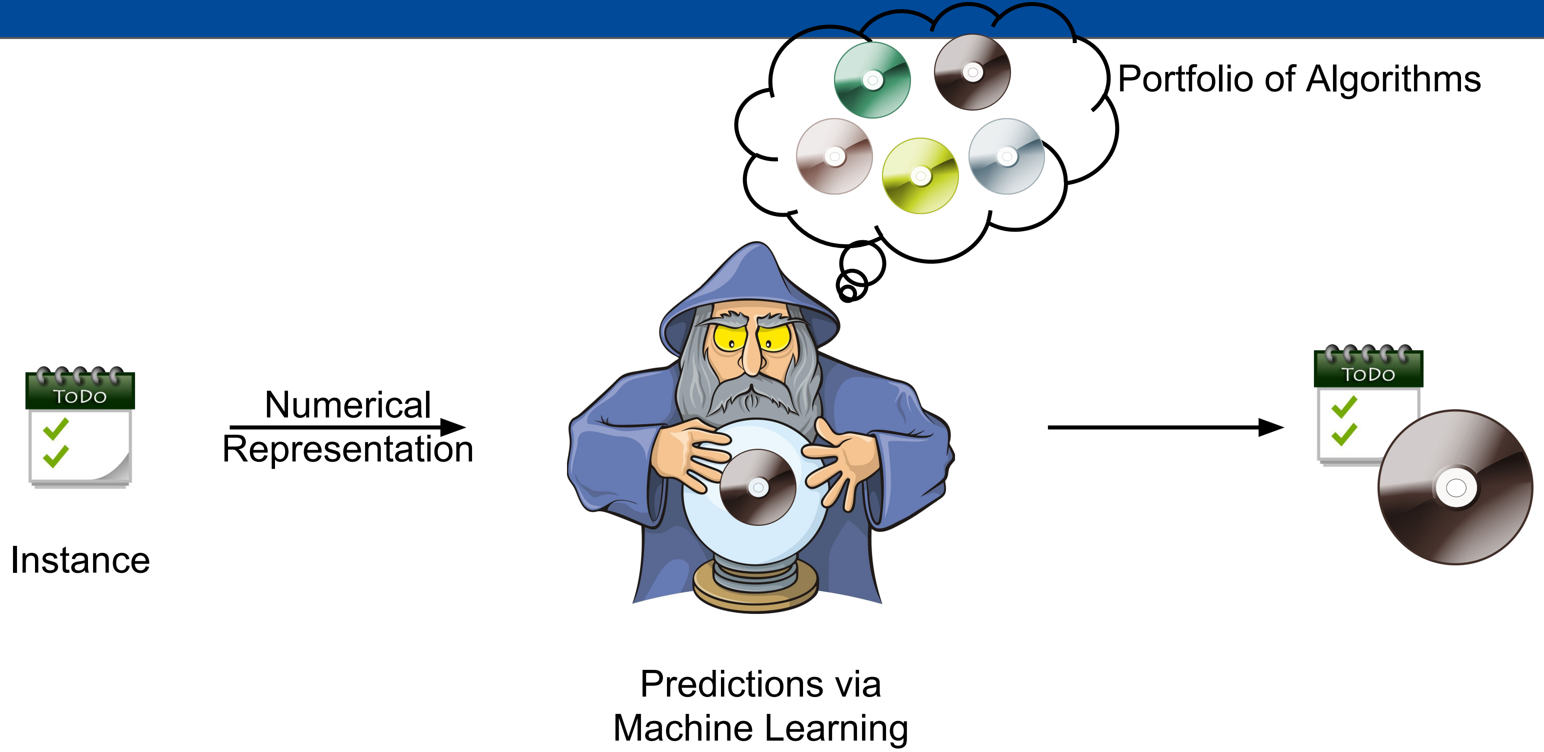
Data available in **ASlib** [Bischl, **Lindauer** et al. AIJ 2016]

Manual Algorithm Selection



Goal: Select the algorithm with the best performance for a given instance

Can we automate Algorithm Selection? [Rice'76]



Goal: *Predict* the algorithm with the best performance for a given instance

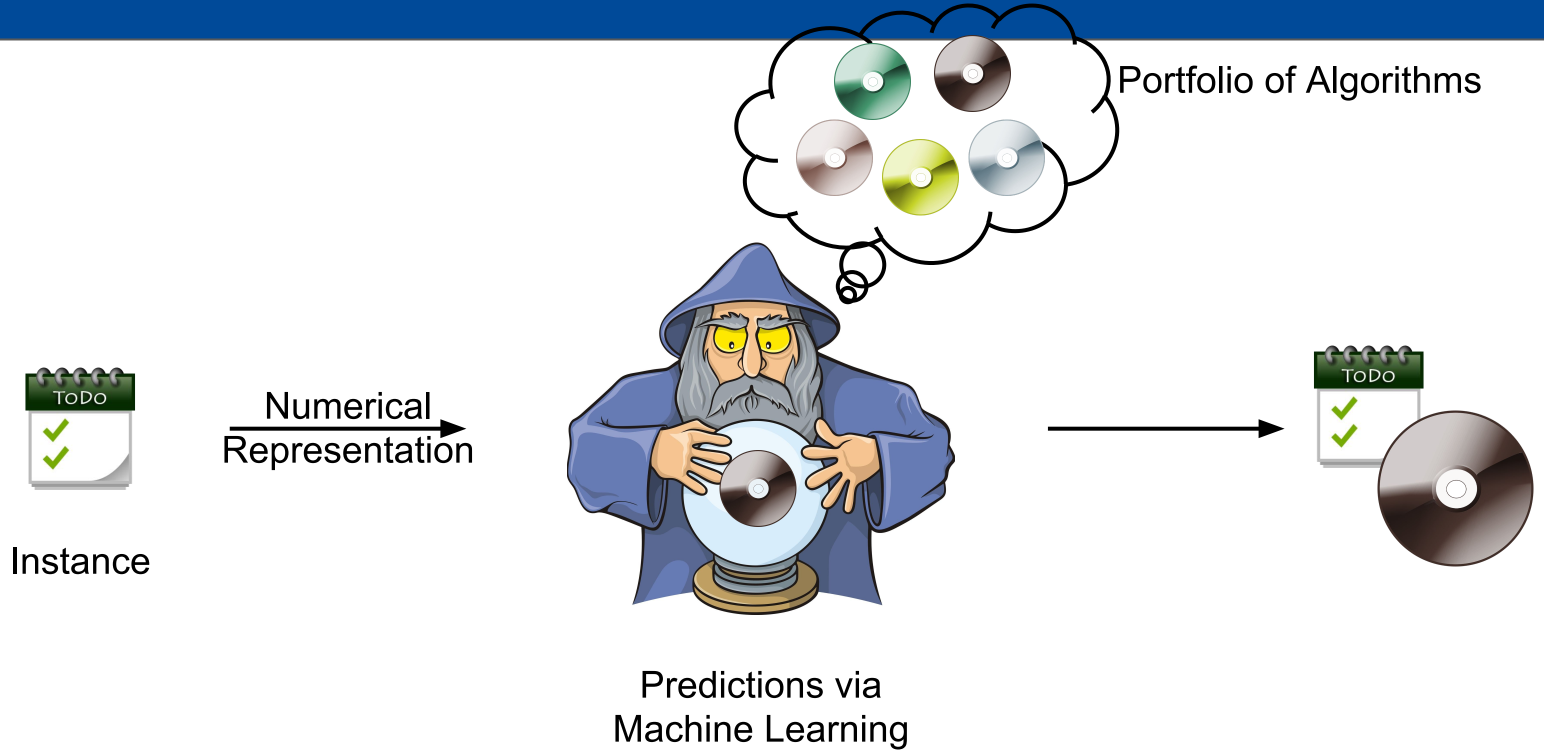
Instance Features = Numerical Representations

- Counting Features
  - How large/hard is the instance?
    - Examples: #variables, #constraints, #data points, #list entries, …

- Probing Features
  - Run a simple algorithm to check behavior
    - Examples: accuracy of decision tree,
      performance of local search SAT solver, …

- Important properties of instance features
  - Informative about performance of algorithms
  - cheap-to-compute

Algorithm Selection [Rice'76]



Goal: *Predict* the algorithm with the best performance for a given instance

Collecting Data → Check out aslib.net

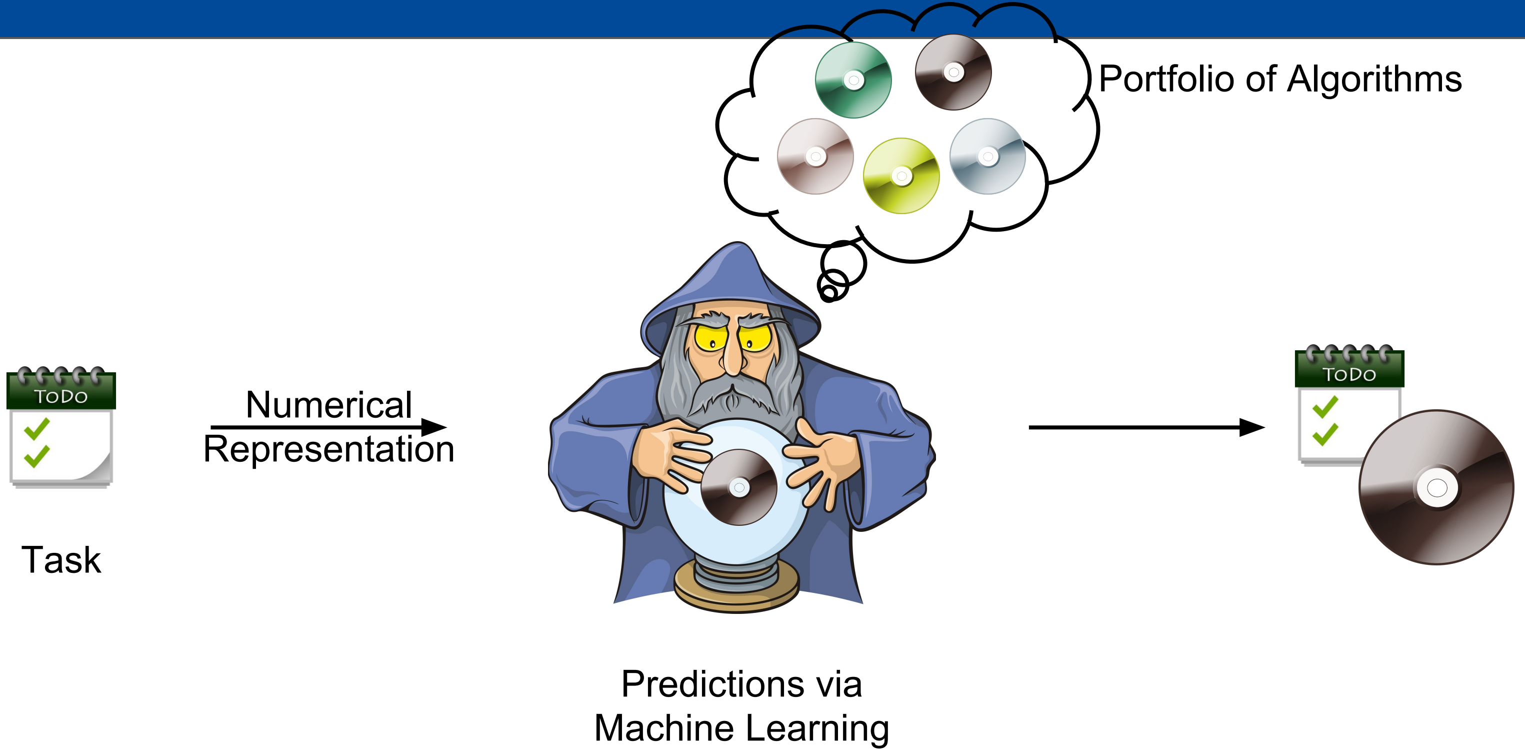
10	**5**	88	10	9
42	24	30	32	**10**
488	888	**55**	487	87
452	123	188	**78**	480
102	488	300	330	**51**
1	18	6	8	88

Training performance data

??	??	??	??	??
??	??	??	??	??
??	??	??	??	??
??	??	??	??	??
??	??	??	??	??
??	??	??	??	??

Unknown test data

Algorithm Selection [Rice'76]



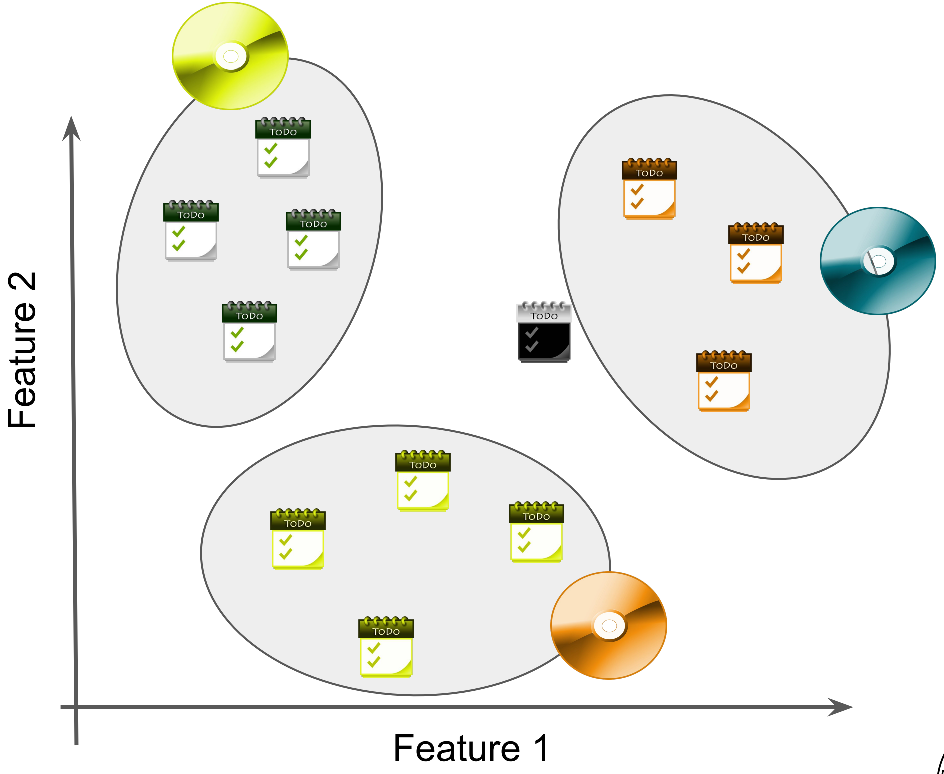
Goal: *Predict* the algorithm with the best performance for a given instance

Approaches

Algorithm Selection: Idea #1 [Kadiolgu et al. 2010]

Idea: Similar instances should be assigned to the same algorithm

- Human-inspired strategy
- 1. cluster instances
- 2. Assign best algorithm in each cluster

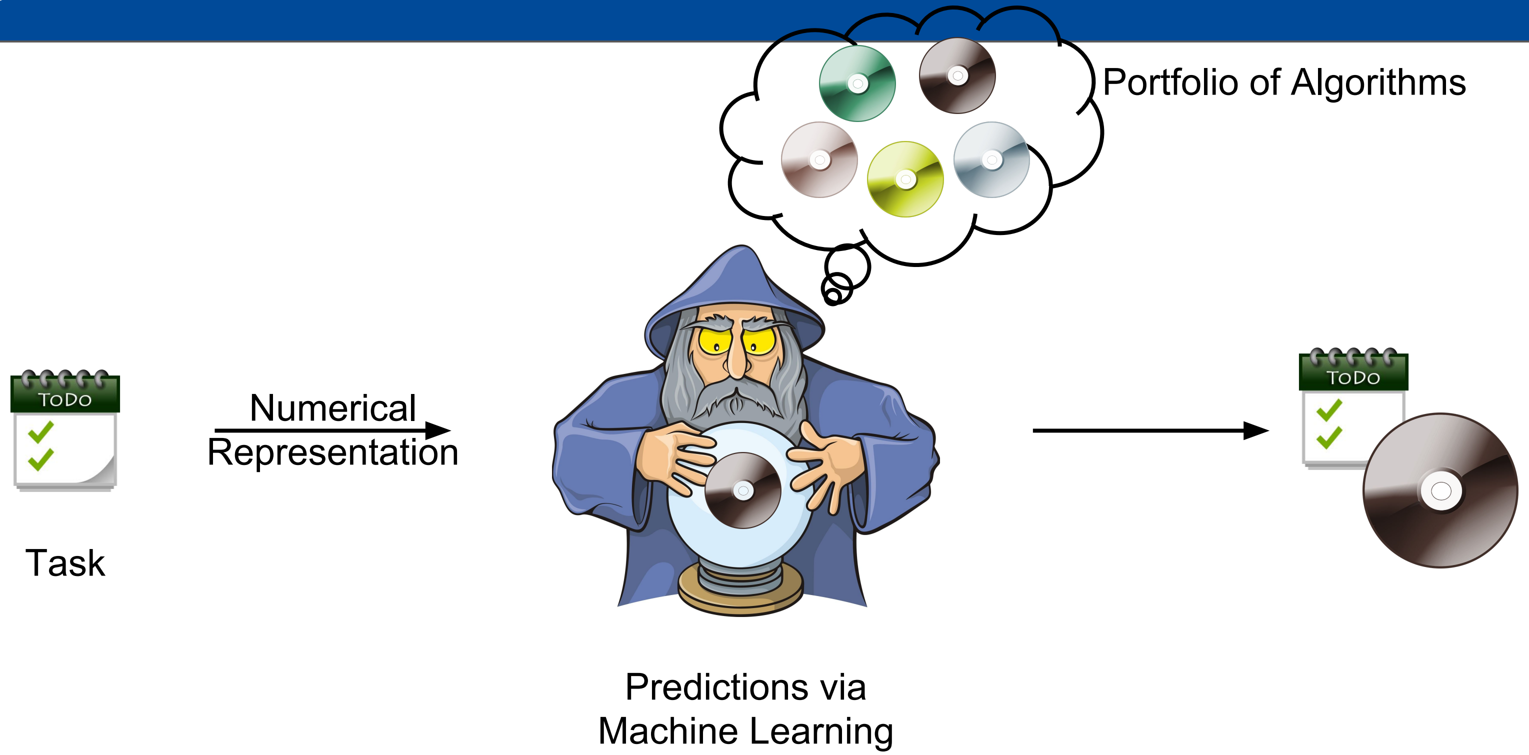


Algorithm Selection: Idea #1 [Kadiolgu et al. 2010]

- Very easy to implement
- Only a single model
- Very fast to train model

- Unsupervised learning
  → clusters could be wrong
- Typically worse performance
  than other approaches

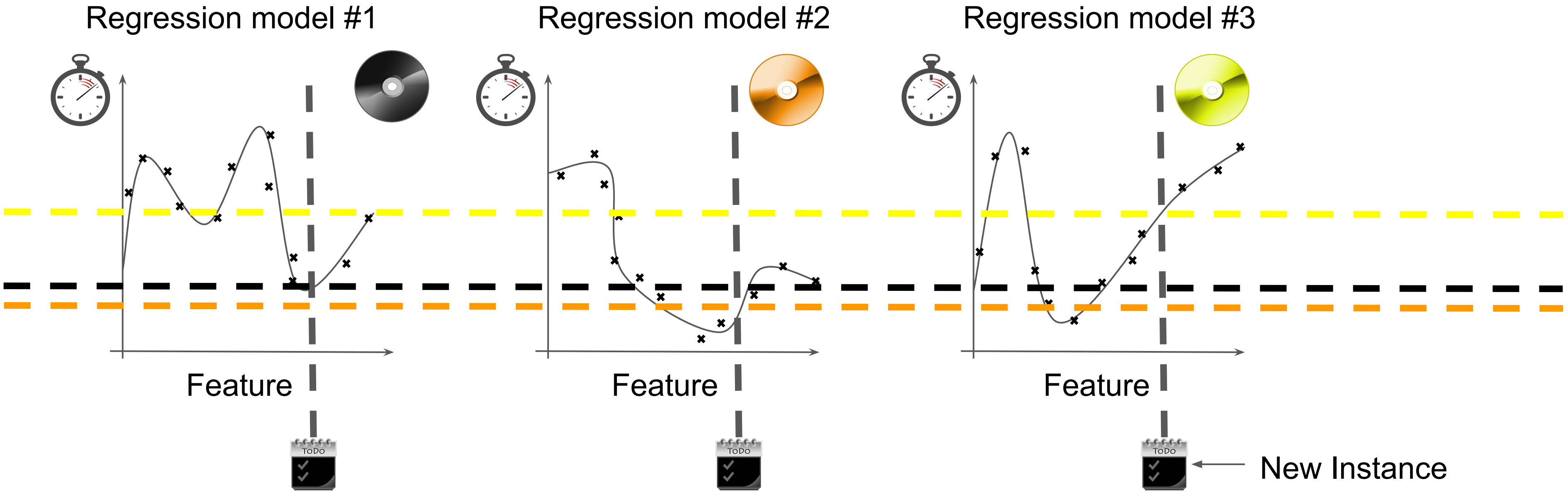
Algorithm Selection [Rice'76]



Goal: *Predict* the algorithm with the best **performance** for a given instance

Algorithm Selection: Idea #2 [Xu et al. 2010]

Idea: Predict the performance of each algorithm and select the best performing one.



Algorithm Selection: Idea #2 [Xu et al. 2010]

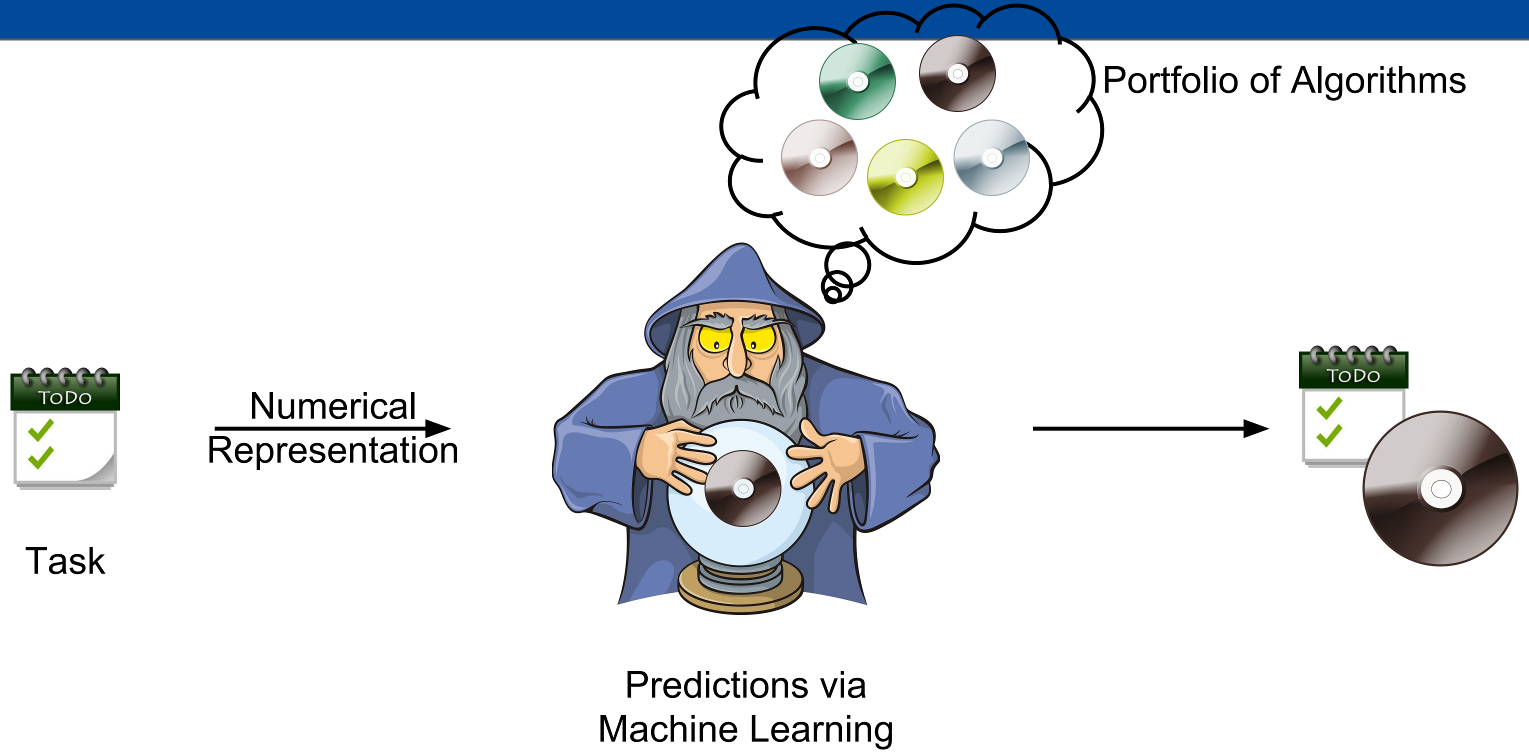
+

- Easy to implement
- Supervised learning
- Can be used for more than algorithm selection

−

- Training of *n* models for *n* algorithms
- Learns a harder task than necessary

Algorithm Selection [Rice'76]



Goal: *Predict* the **algorithm** with the best performance for a given instance

Algorithm Selection: Idea #3 [Xu et al. 2011]

Idea: Learn a classification model for each pair of algorithms

			Label	Weight
	10	**5**	1	5
	42	**24**	1	18
	488	888	0	400
	452	**123**	1	329
	102	488	0	386
	1	18	0	17

Less important instances

Very important instances

Algorithm Selection: Idea #3 [Xu et al. 2011]

Idea: For a new instance, use a voting scheme on pairwise predictions

--	0	0	1	0
1	--	1	1	1
1	0	--	1	0
0	0	0	--	0
1	0	1	1	--

#Votes	
1	
4	← Selected algorithm
2	
0	
3	

Algorithm Selection: Idea #3 [Xu et al. 2010]

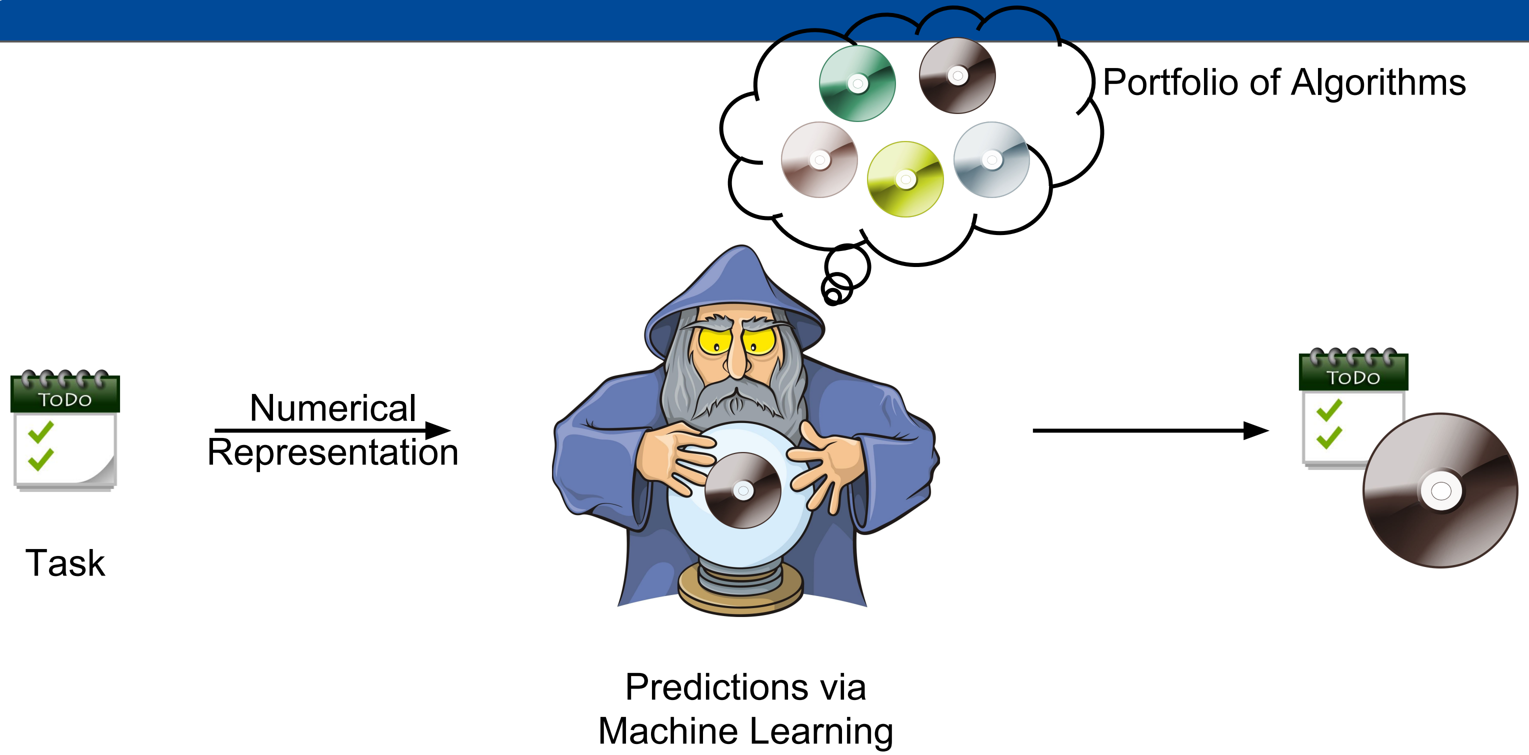
+

- Supervised learning
- Weighting of instances
- State-of-the-art approach

−

- Training of $n^2/2$ models for n algorithms
- Predictions from $n^2/2$ models

Algorithm Selection [Rice'76]



Goal: *Predict* the **algorithm with the best performance** for a given instance

Algorithm Selection: Idea #4 [Kotthoff 2015]

Idea: Learn pairwise regressions model for difference in performance

10	**5**
42	24
488	888
452	123
102	488
1	18

Performance difference
5
18
-400
329
-386
-17

Algorithm Selection: Idea #4 [Kotthoff 2015]

Idea: For a new instance, sum up differences in performance predictions

--	-2	-5	-100	50
2	--	1	11	5
5	-1	--	100	-5
100	-11	-100	--	-12
-50	-5	5	12	--

Advantage	
-57	
19	← Most Votes
99	← Selected algorithm
-23	
-38	

Algorithm Selection: Idea #4 [Kotthoff 2015]

+

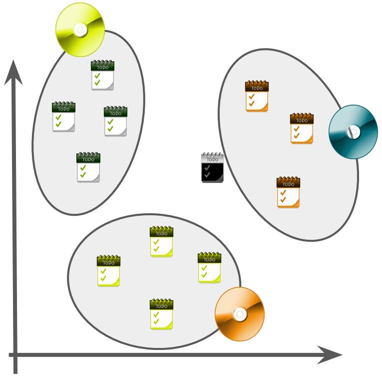
- Supervised learning
- Takes performance difference in labels into account

−

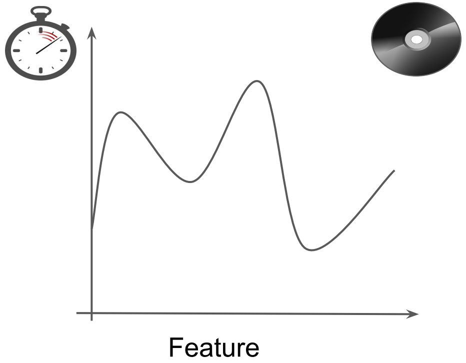
- Training of $n^2/2$ models for n algorithms
- Predictions from $n^2/2$ models

Overview of Algorithm Selection Approaches

Clustering



Regression



Pairwise Classification

--	0	0	1	0
1	--	1	1	1
1	0	--	1	0
0	0	0	--	0
1	0	1	1	--

Predictions

Pairwise Regression

--	-2	-5	-100	50
2	--	1	11	5
5	-1	--	100	-5
100	-11	-100	--	-12
-50	-5	5	12	--

Predictions

Automate Automated Algorithm Selection

Comparison of Algorithm Selection Approaches

Applications

Selection Tools

	3S-like	aspeed	claspfolio-1.0-like	ISAC-like	ME-ASP-like	SATzilla'09-like	SATzilla'11-like
ASP-POTASSCO	4.1	1.4	2.8	3.8	1.9	2.9	4.2
CSP-2010	1.5	1.0	2.1	2.1	2.6	2.5	3.1
MAXSAT12-PMS	6.5	2.7	1.6	4.9	2.1	3.4	8.6
PREMARSHALLING	2.9	3.6	1.2	1.3	1.1	1.5	2.3
PROTEUS-2014	10.9	6.3	3.5	4.3	3.1	4.9	6.5
QBF-2011	7.7	4.9	2.3	2.8	2.8	3.7	9.8
SAT11-HAND	2.6	3.6	1.1	1.2	1.0	1.9	2.3
SAT11-INDU	1.2	1.1	1.2	1.3	1.2	1.1	1.2
SAT11-RAND	3.9	4.7	1.2	2.5	1.8	2.6	3.8
SAT12-ALL	1.5	1.1	1.2	1.1	1.1	1.4	1.8
SAT12-HAND	1.7	1.8	1.1	1.1	1.0	1.5	1.9
SAT12-INDU	1.2	0.8	1.2	1.2	1.1	1.3	1.3
SAT12-RAND	0.8	0.8	0.6	0.9	0.9	0.9	0.9
geo. mean	2.6	2.0	1.5	1.9	1.5	2.0	2.8

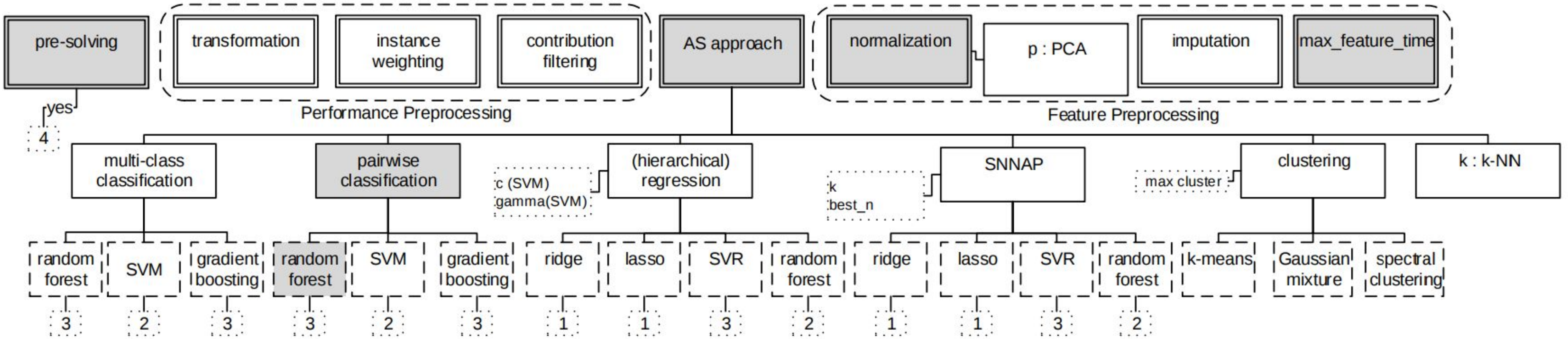
Insight:
Different applications require different selection approaches!

Challenges in Applying Algorithm Selection

- For each application, we potentially need a different approach
  - clustering vs. regression vs. pairwise classification vs. pairwise regression
- Each approach can be implemented with different machine learning algorithms
  - Random forest, SVM, deep neural network, gradient boosting
- Each machine learning algorithm requires optimal hyperparameter settings
  - Kernel width of SVM?
  - Pruning strength of trees?
  - …

→ *Effective application of algorithm selection in practice can be hard!*

Algorithm Selection Design Choices

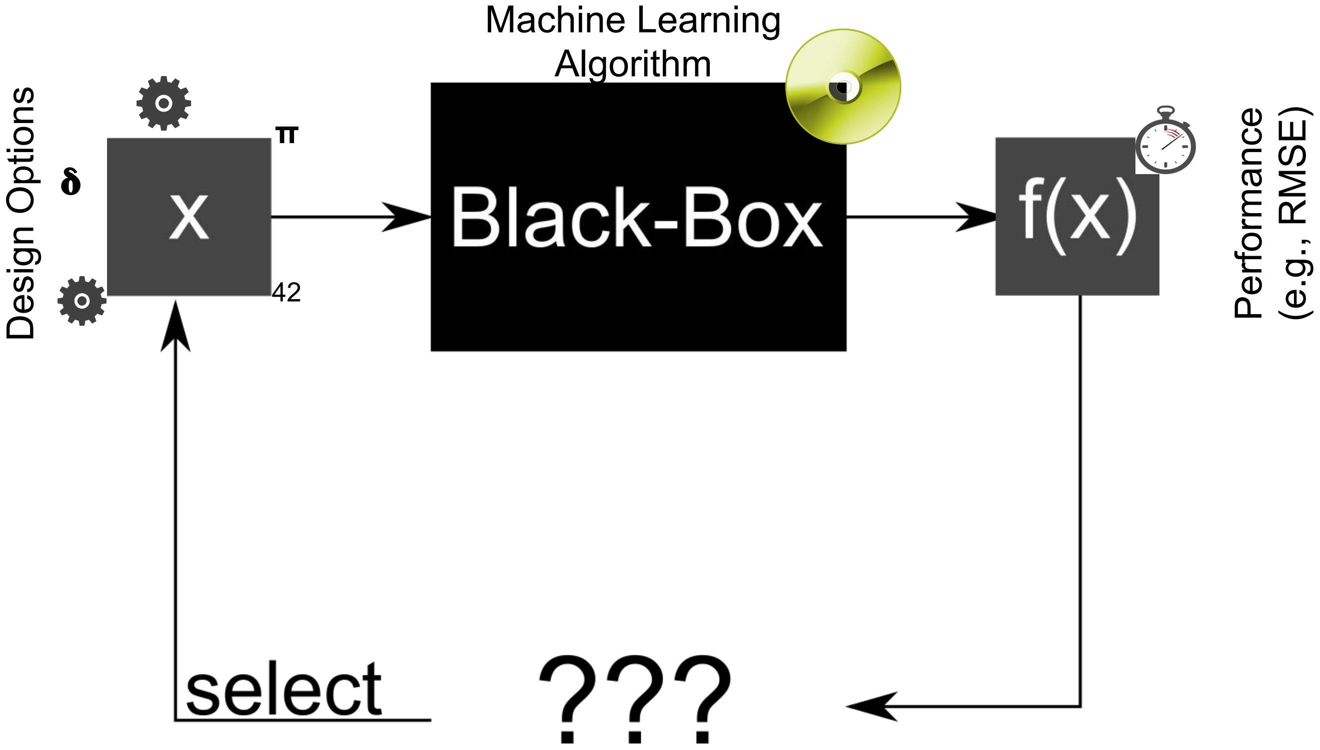


AutoML saves the day!

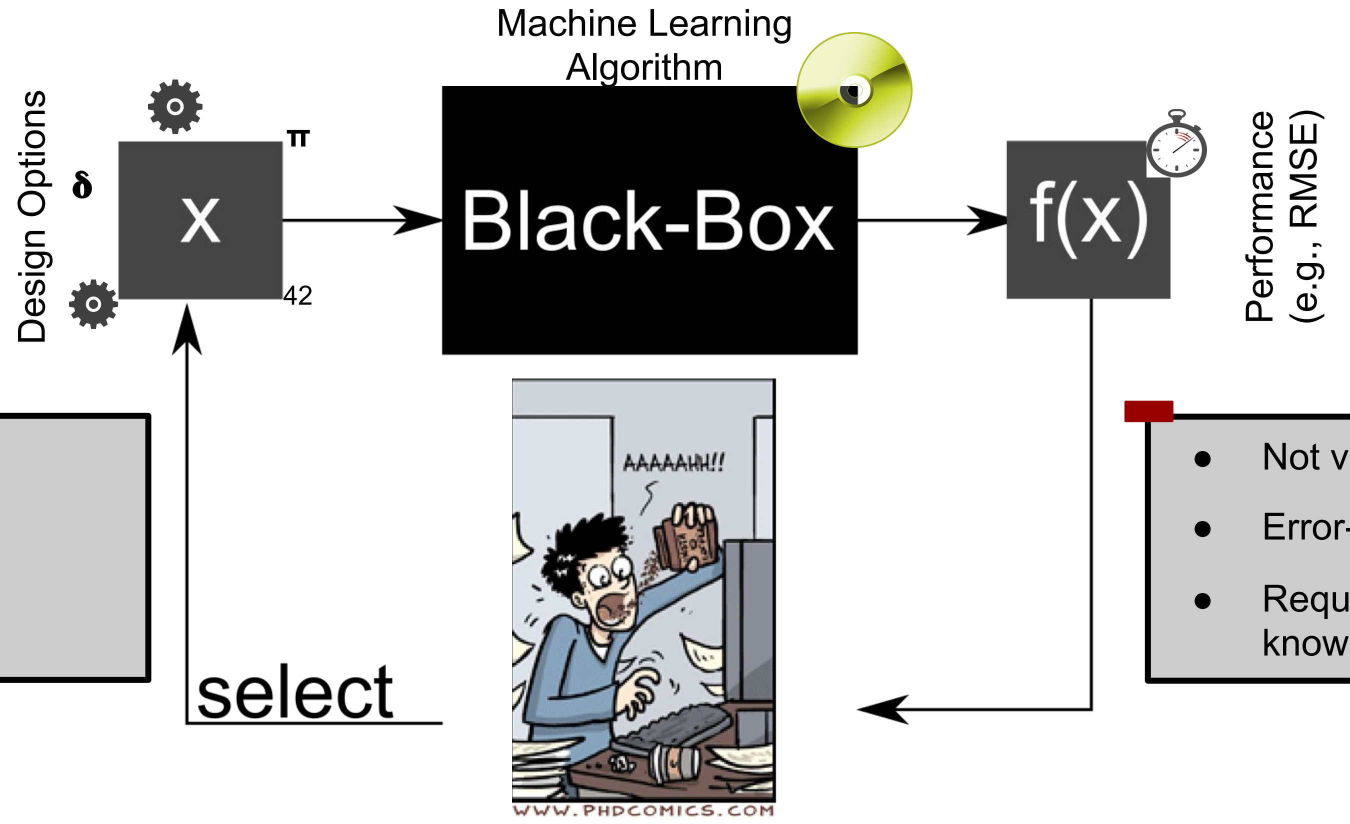
- Insight: Algorithm selection is yet another machine learning problem (with a special design spaces)

- Automated machine learning:
  - Automated search for best machine learning algorithm and its hyperparameter settings
  - Allows for automated deployment of algorithm selection in practice

Crash course: AutoML



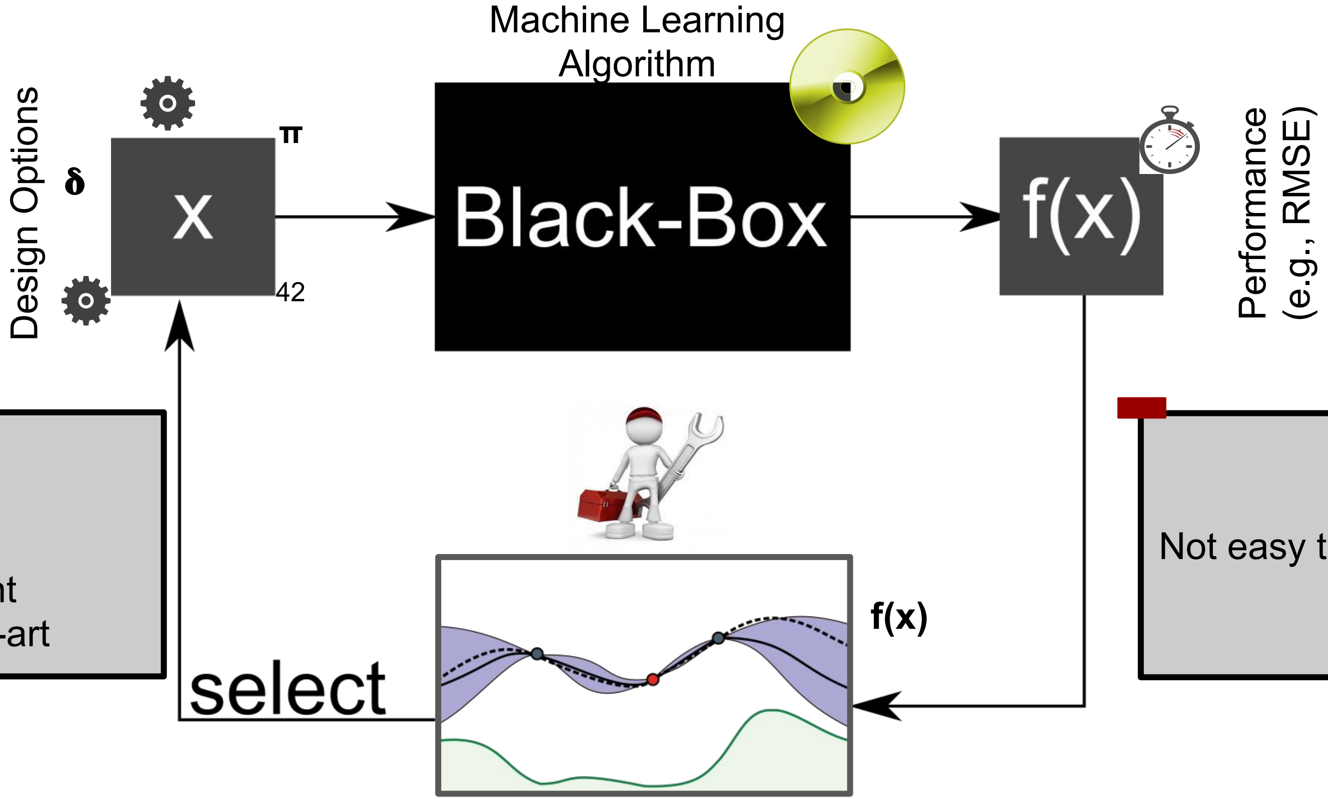
Crash course: AutoML



...

- Not very efficient
- Error-prone
- Requires expert knowledge

Crash course: AutoML

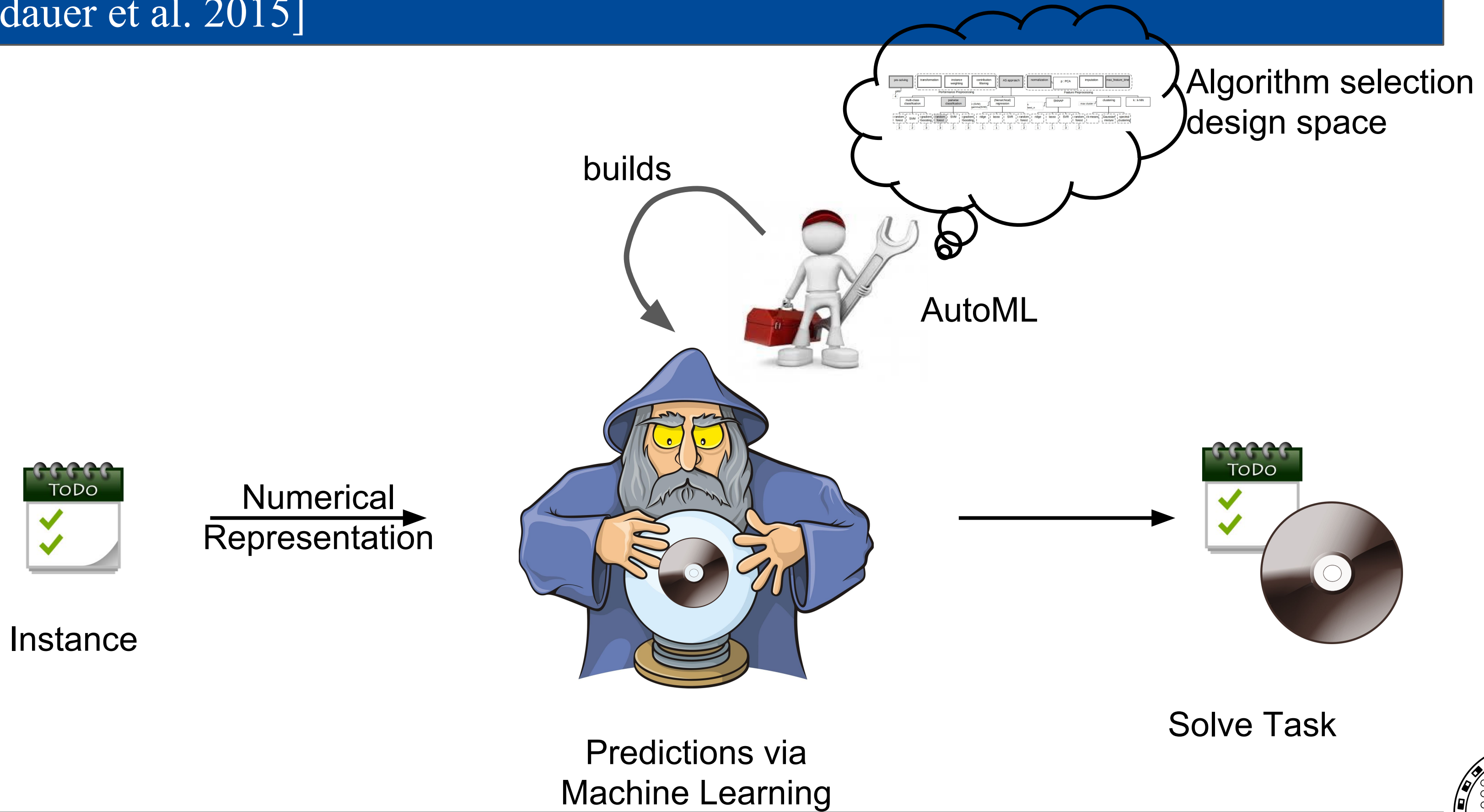


- Trade-off Exploration-Exploitation
- Data efficient
- State-of-the-art

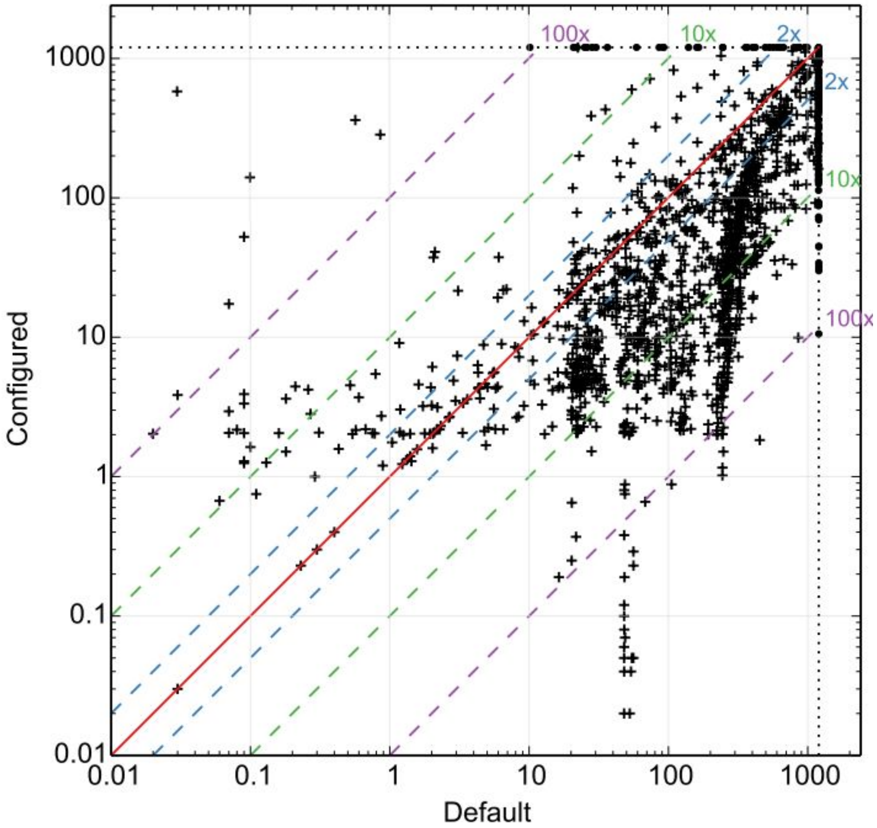
Not easy to parallelize

AutoFolio: Algorithm Selection + AutoML

[Lindauer et al. 2015]



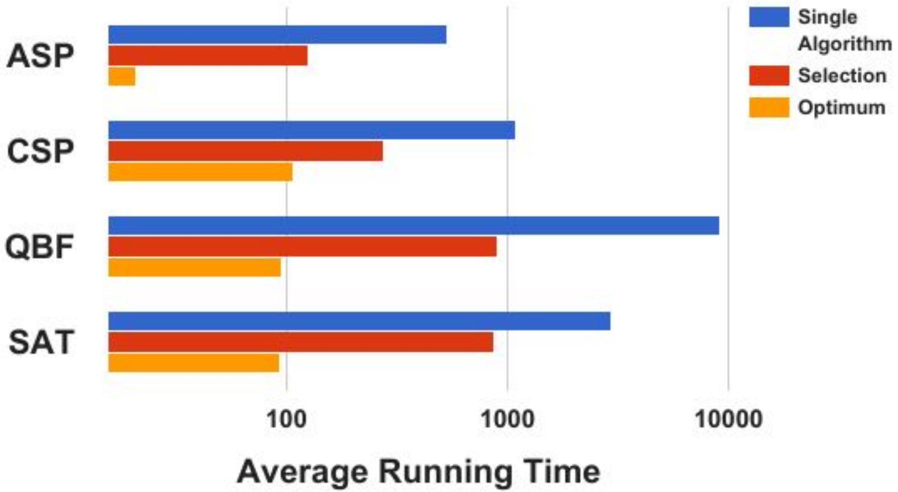
AutoFolio on SAT [Lindauer et al. 2015]

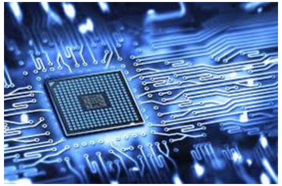


Legend:

- Dot: one instance
- Metric: runtime
- Portfolio: set of SAT solvers
- x-axis: default selection approach
- y-axis: optimized selection approach

AutoFolio [Lindauer et al. 2015]



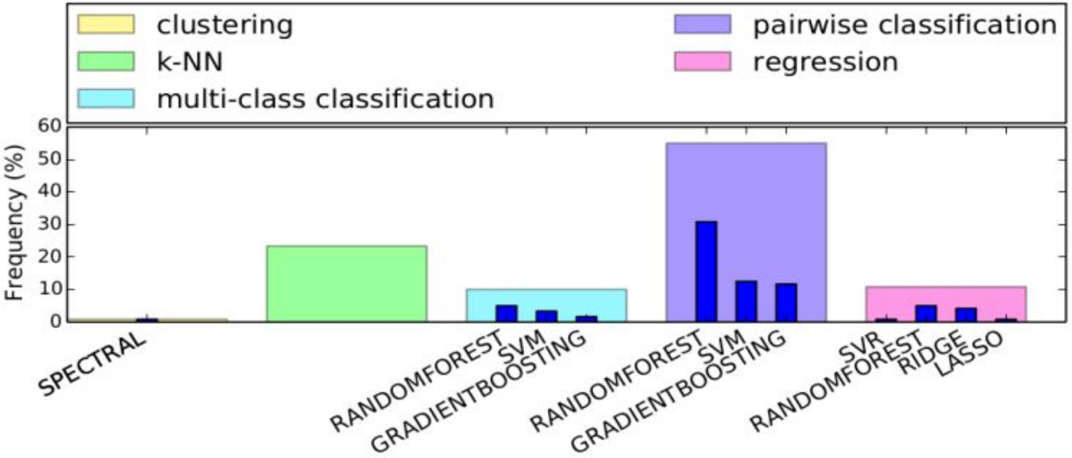


5 - 10 fold speedup!

(on these examples)

Insights from AutoFolio [Lindauer et al. 2015]

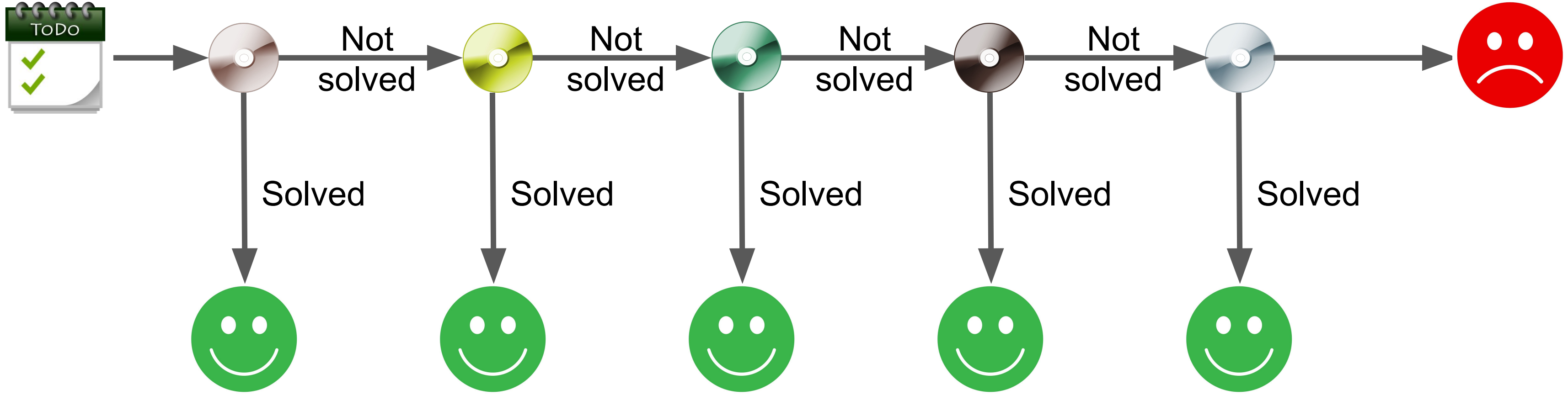
- Most important design decision:
  How much time do I invest in instance feature computation?
- 2nd most important design decision:
  Algorithm selection approach



Extensions of Algorithm Selection

Schedules of Algorithms

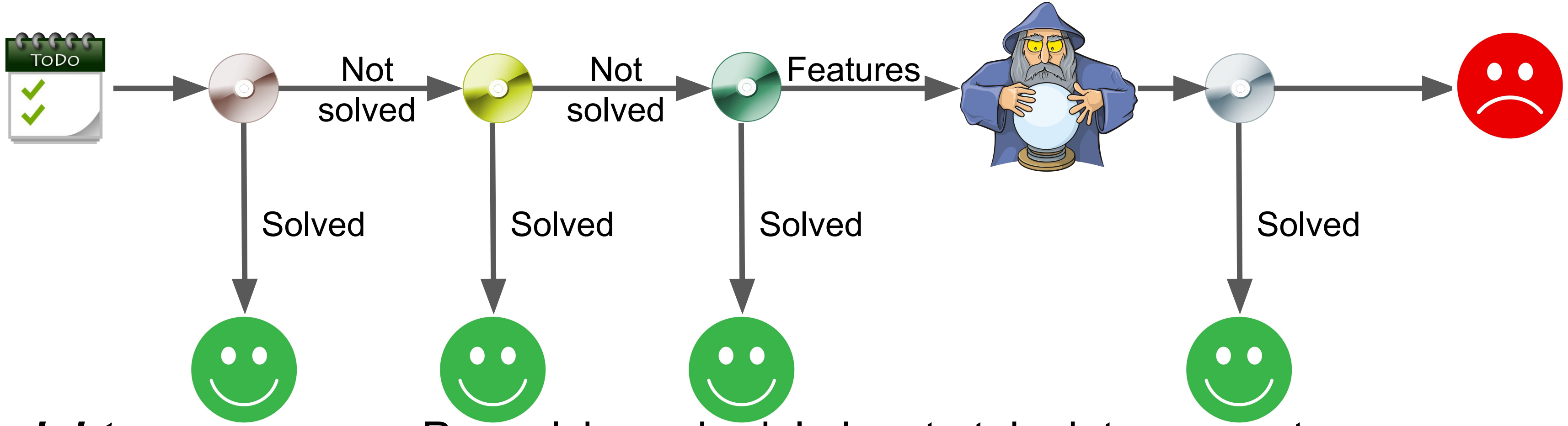
Idea: Instead of a single algorithm,
we want to run several algorithms in a sequence



Idea #1: Pre-solving schedules [Xu et al. 2010]

Idea: First runs a static schedule of algorithms (independent of given instance).
If it fails, use algorithm selection (based on instance features).

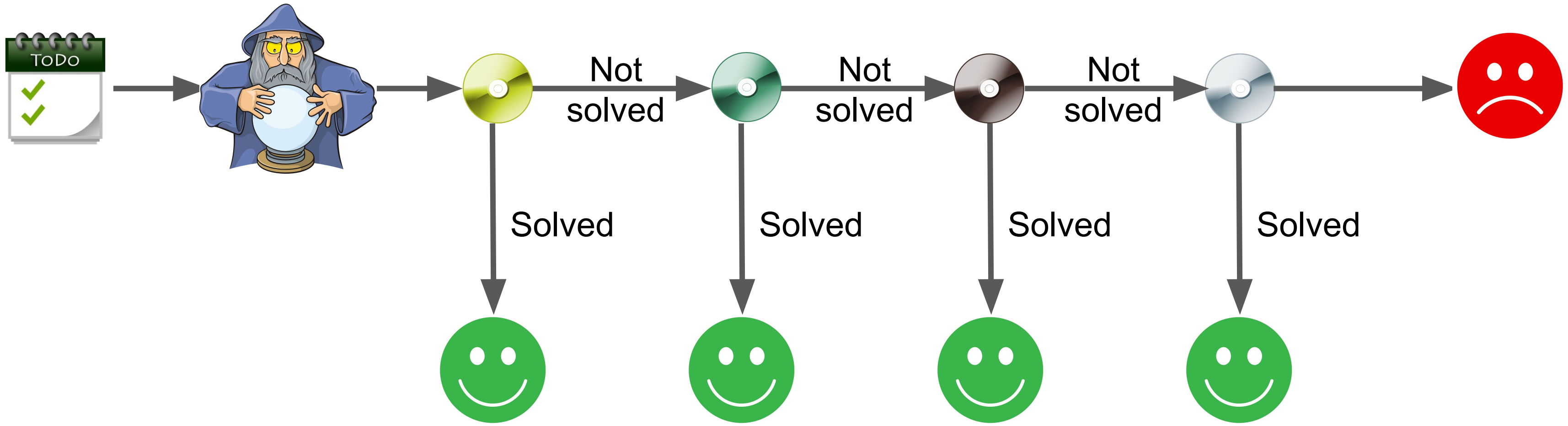
Challenge: Find a well-performing schedule → hard optimization-problem



Insight [Gonard et al. 2016] **:** Pre-solving schedule has to take into account prediction model and vice versa.

Idea #2: Predict schedule of algorithms

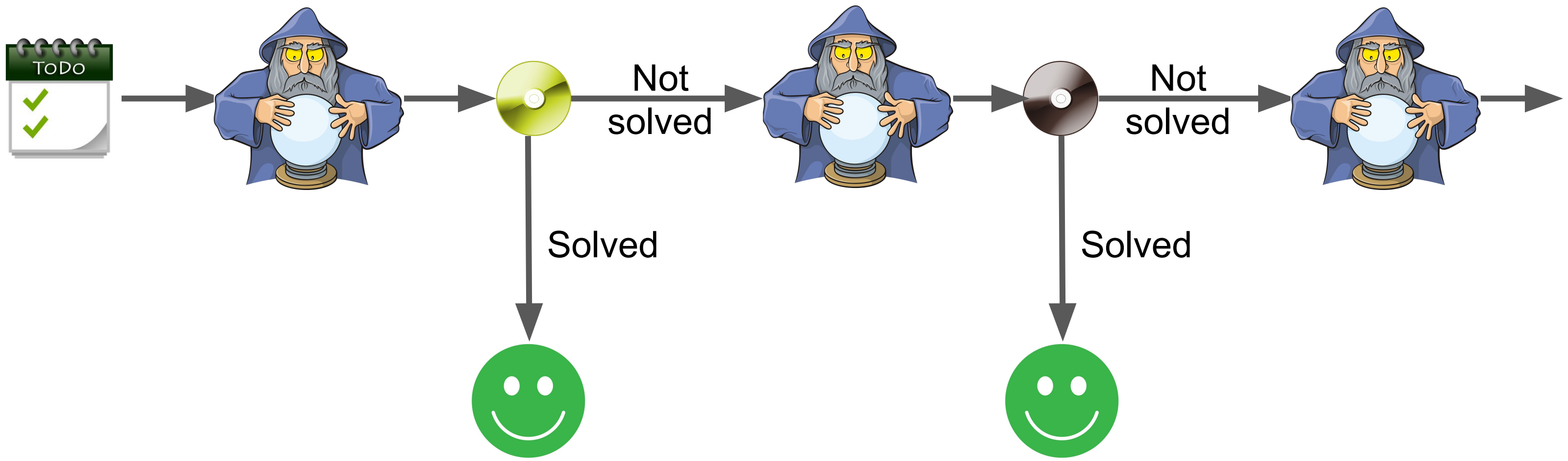
Idea: Predict a schedule of algorithms once in the beginning.
The schedule is instance-dependent.



Idea [Amadini et al. 2014]: Sort algorithms by probability to solve given instance.

Idea #3: Sequential Predictions

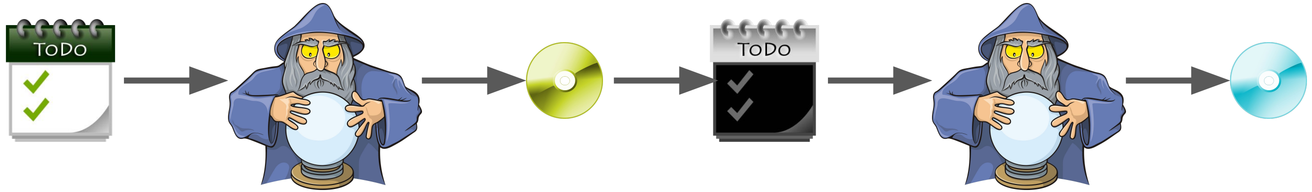
Idea: Predict an algorithm and update your believe based on information of previous algorithm runs.



Online Algorithm Selection

Problem: Assumption of basic ML is
that trainings distributions representative of test distribution.

The Truth: Concept drifts are quite likely, i.e., training is not representative of test.



Idea: Each solved instance corresponds to new knowledge
and the model can be updated.

Challenge: We need exploration to update our models.

Online Algorithm Selection as a Bandit Problem
[Degroote et al. 2016]

Idea:

- Each algorithm is a bandit.
- For each instance, we have to decide whether
    1. to exploit our current belief (model )
    2. To explore algorithms
- Can be modelled with upper confidence bounds (UCB)
- Greedy policy is a surprisingly strong baseline [Degroote et al. 2016]

End-to-End Algorithm Selection

Insight: Most important part is the design of instance features.

Instance

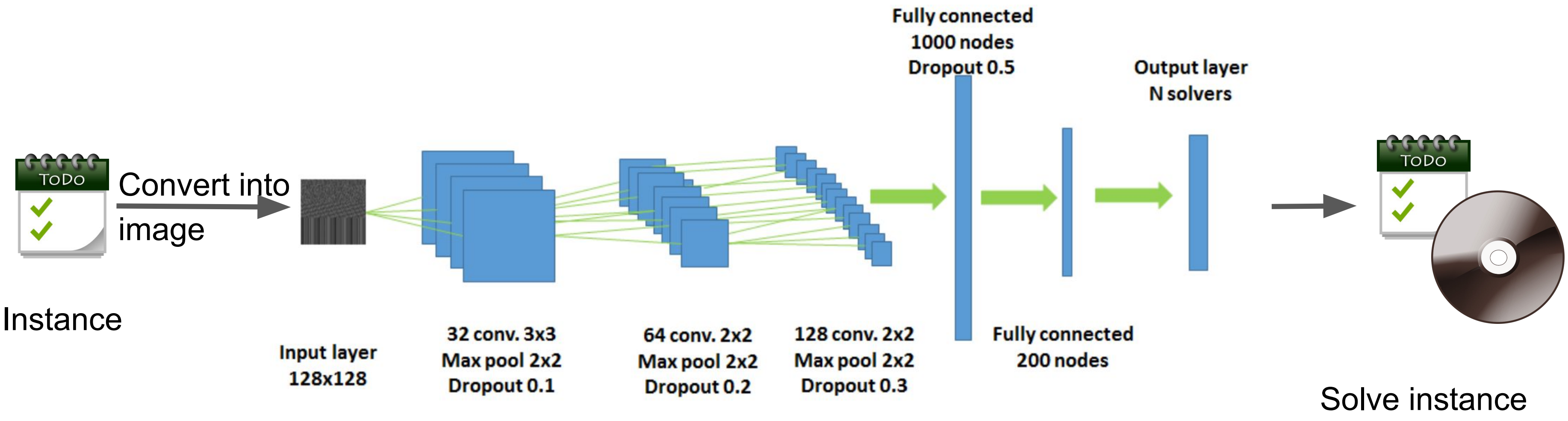
Numerical Representation

Solve instance

Idea: Replace hand-designed features by a neural network

Deep Learning for Algorithm Selection [Loreggia et al. 2016]

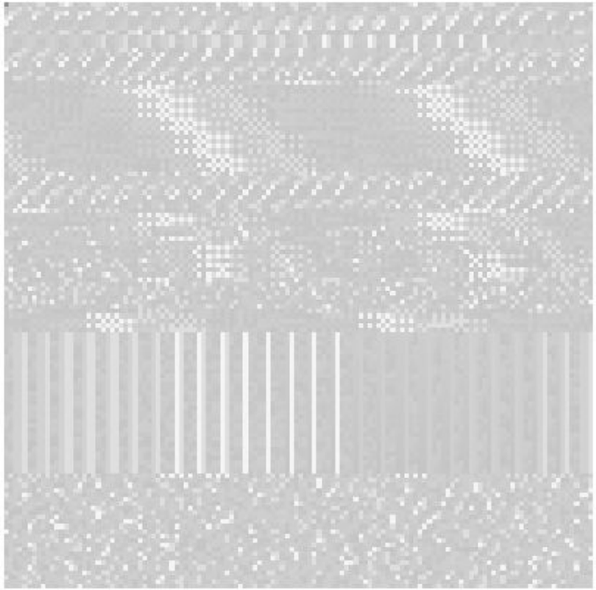
Idea: Replace hand-designed features by a neural network



Instances as Images

1. Each character translated into ASCII
   - ASCII can be seen as grayscale encoding
2. For *n* characters in file, reshape into square root(*n*) x square root(*n*)
3. Compress into 128x128 pixels
4. Use CNN to classify image

→ works fairly well, but worse than expert features



SAT instance as image

Open Challenges in Algorithm Selection

Open Challenges

1. Generic way for **generating high-quality features**
   - For some domains, we still don’t know good features
   - Deep learning for instance features is still not mature

2. Efficient use of **life-long learning**?
   - Greedy online algorithm selection can’t be the final answer

3. Algorithm selection for **multi-core systems**
   - How to balance exploitation and exploration
     if we can select *k* out of *n* algorithms?

Thank you!