Automatic Online Algorithm Selection

Hans Degroote
KU Leuven
Supervised by Patrick De Causmaecker

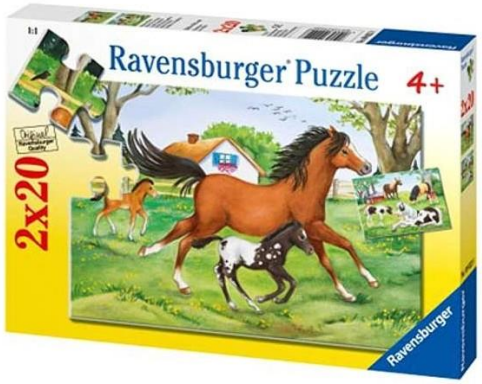
Hans.Degroote@kuleuven.be

What To Expect

- **Automatic Algorithm Selection**
- Automatic Online Algorithm Selection
- Exploration vs. Exploitation and Multi-armed Bandits
- Exploration vs. Exploitation in Automatic Online Algorithm Selection

Automatic Algorithm Selection (metaphor)

- General problem (solving puzzles)
- Specific problem instances (actual puzzles)



- Problem features
  - Amount of pieces
  - Puzzle content (animals, abstract, landscape…)
  - Piece similarity
  - …

Automatic Algorithm Selection (metaphor)

- Different puzzlers are available
  - Complementary: different puzzlers are fastest on different puzzles

- Problem statement

given a new puzzle, solve it using the best puzzler for it

Automatic Algorithm Selection (metaphor)

- Solution strategy
  - Collect training puzzles and make each puzzler solve them
  - Measure their runtime
  - Create a classification model based on puzzle features

  - For a new puzzle:
    - Calculate feature values
    - Solve the puzzle using the puzzler predicted by the classification model

  - Still to decide: how to measure performance?

Automatic Algoritm Selection (metaphor)

- What if a puzzler takes forever?
  - Cap the runtimes

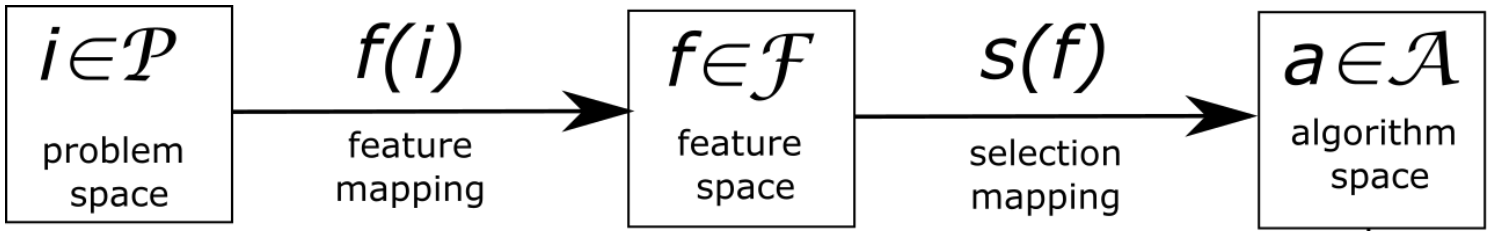
- Performance measure: Penalised runtime (PAR)
  - Time-limit of 24 hours
  - Puzzle solved within time limit => runtime
  - Time-out => 24 hours* 10 (PAR10)

Automatic Algoritm Selection (metaphor)

Training data

>24h	20h	10h
2 min	10 min	45 min
120 min	30 min	60 min
110 min	65 min	30 min

Automatic Algorithm Selection



Optimise $\mathbf{E}_Q[(p(s(f(i)),i)]$

With Q = probability distribution over problem space

Automatic algorithm selection is a cost-sensitive classification problem

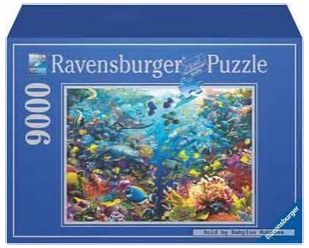
-Rice, John R. "The algorithm selection problem." Advances in computers 15 (1976): 65-118.

Cost-sensitive classification

- Cost-sensitive classification examples
  - Is this an image of a malignant cancer?
    - Cost of false positive: additional tests (+ some fear)
    - Cost of false negative: …
  - Is this $1 million credit card transaction legitimate?
    - Cost of false positive: $1 million + an angry/worried phone call from the client
    - Cost of false negative: an angry phone call from the client

Elkan, Charles. "The foundations of cost-sensitive learning." International joint conference on artificial intelligence. Vol. 17. No. 1. LAWRENCE ERLBAUM ASSOCIATES LTD, 2001.

Cost-sensitive classification problem with example-specific costs



>24 hours

6 hours

3 hours

10 hours

119 minutes

120 minutes

-Bischl, Bernd, et al. "Algorithm selection based on exploratory landscape analysis and cost-sensitive learning." Proceedings of the 14th annual conference on Genetic and evolutionary computation. ACM, 2012.

Work Related to Automatic Algorithm Selection

- Meta-learning
  - Synonym, used in the machine-learning community
    - Their algorithms are classifiers
    - Their problem space is the space of all classification problems
- Automated parameter tuning/Automatic algorithm configuration
- Parallel portfolios of algorithms

Automated Parameter Tuning/Automatic Algorithm Configuration

- Useful for parametrised algorithms
- Learning the optimal parameter configuration of an algorithm for a specific instance distribution
  - Often a search over an infinite amount of parameter-combinations
  - In the general case the same parameter configuration is used for all instances
- Example methods
  - Irace
  - SMAC

-López-Ibánez, Manuel, et al. The irace package, iterated race for automatic algorithm configuration. Technical Report TR/IRIDIA/2011-004, IRIDIA, Université Libre de Bruxelles, Belgium, 2011.
-Hutter, Frank, Holger H. Hoos, and Kevin Leyton-Brown. "Sequential model-based optimization for general algorithm configuration." International Conference on Learning and Intelligent Optimization. Springer Berlin Heidelberg, 2011.

Static Parallel Portfolios

- Used for decision problems
  - Goal: find a solution as fast as possible
- Learn the optimal allocation of time shares to algorithms
  - Example schedule: Alg1-30%, Alg2-40%, Alg3-30%, Alg4-0%
- The same time-allocation is used for all instances

Petrik, Marek, and Shlomo Zilberstein. "Learning parallel portfolios of algorithms." Annals of Mathematics and Artificial Intelligence 48.1-2 (2006): 85-106.

Automatic Algorithm Selection: Methods (some examples)

- Regression
  - Empirical hardness models
- K-Nearest Neighbours
- Pairwise Decision Forest

- Survey of methods anno 2009:

  -Smith-Miles, Kate A. "Cross-disciplinary perspectives on meta-learning for algorithm selection." ACM Computing Surveys (CSUR) 41.1 (2009): 6.

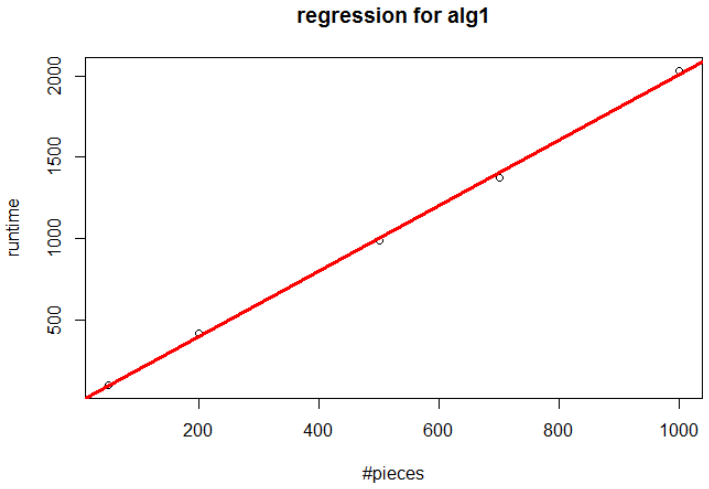
Regression/Empirical Hardness Models

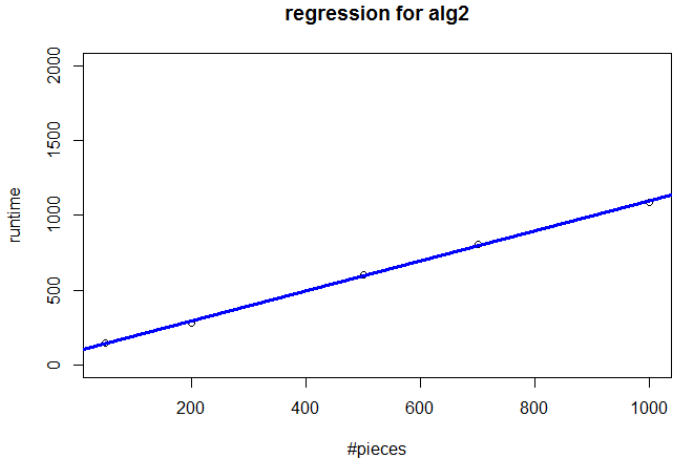
- Create a model for each algorithm predicting its performance in function of instance features

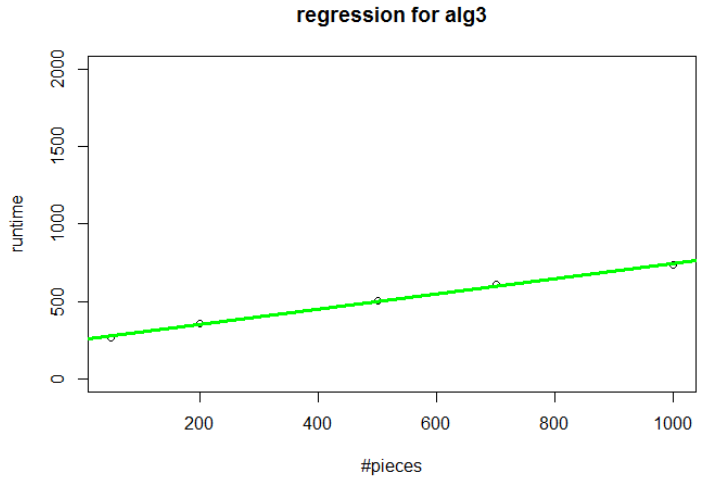
- Used successfully for the Satisfiability Problem (SATzilla)
  - Dominated the SAT-competitions
    - until portfolio approaches were banned from the regular track

Xu, Lin, et al. "SATzilla: portfolio-based algorithm selection for SAT." Journal of Artificial Intelligence Research 32 (2008): 565-606.

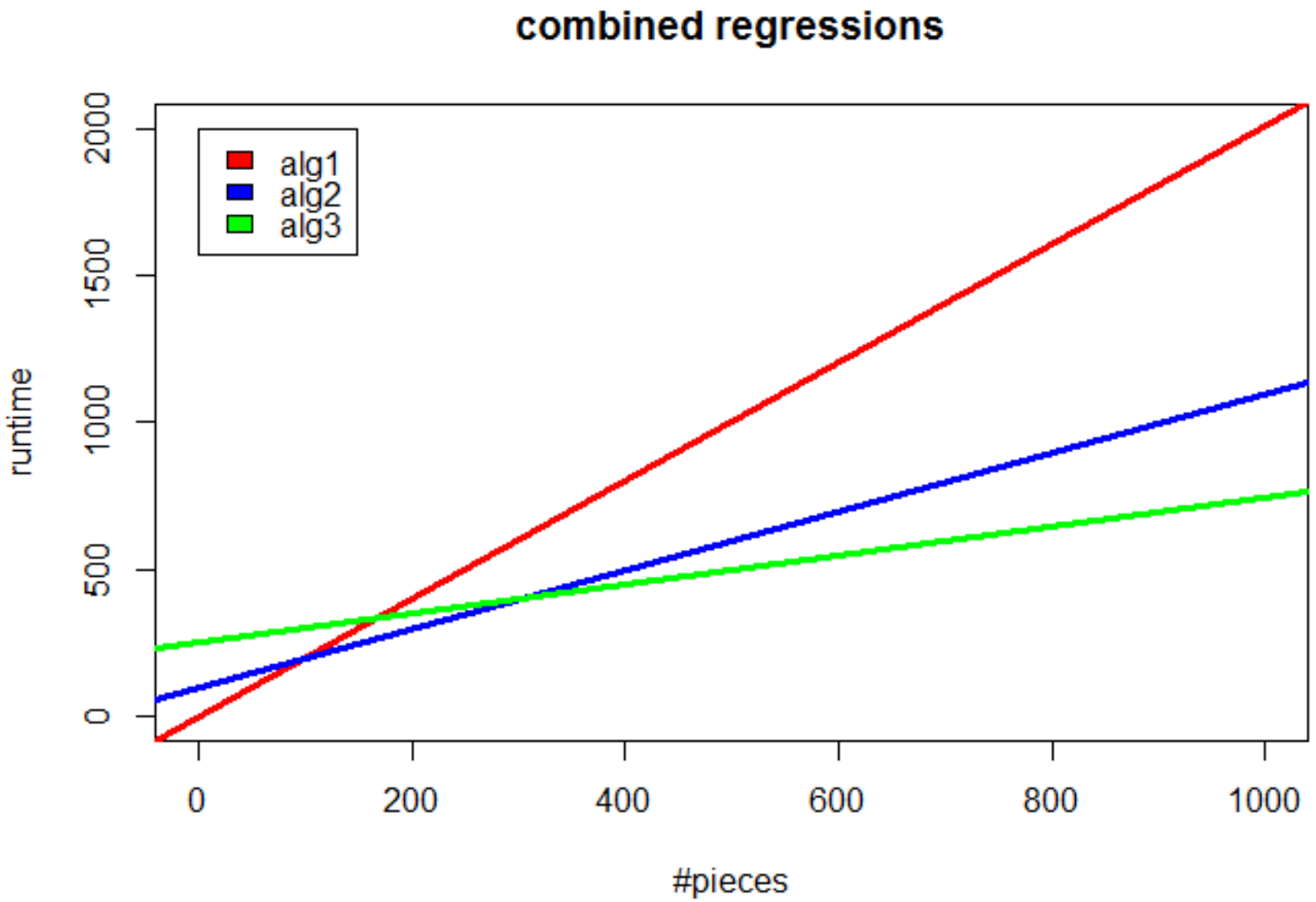
Regression-based Algorithm Selection (metaphor)







Regression-based Algorithm Selection (metaphor)



K-Nearest Neighbours

- Normalise all features to values between 0 and 1
- Define a similarity measure between points in the feature space
  - Euclidian distance

- For a new instance
  - Identify the k nearest neighbours
  - Check which algorithm performs best on those instances
  - Select it

-Kadioglu, Serdar, et al. "Algorithm selection and scheduling." International Conference on Principles and Practice of Constraint Programming. Springer Berlin Heidelberg, 2011.

Pairwise Decision Forest

- Create a decision forest classifier for every pair of algorithms
  - Predicting which of the two is best
- Select the algorithm most often predicted to be best

- Successfully used in the new version of SATzilla
- Considered by some to be the current state of the art method

-Xu, Lin, et al. "SATzilla2012: Improved algorithm selection based on cost-sensitive classification models." Proceedings of SAT Challenge (2012): 57-58.

Automatic Algorithm Selection Methodology

- 2 main kinds of solution strategies
  - Direct classification
    - Instance features -> algorithm
  - Intermediary step using performance (regression)
    - Create a regression model per algorithm
    - Instance features -> performance prediction for each algorithm -> algorithm

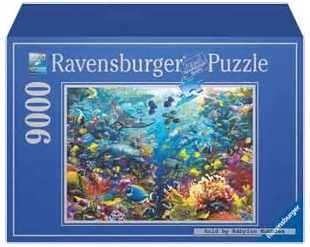
What To Expect

- Automatic Algorithm Selection
- **Automatic Online Algorithm Selection**
- Exploration vs. Exploitation and Multi-armed Bandits
- Exploration vs. Exploitation in Automatic Online Algorithm Selection

Automatic Online Algorithm Selection (metaphor)

Our puzzlers are used to solve new puzzles for which they are predicted to be best, generating new performance data

+



> 24 h

+

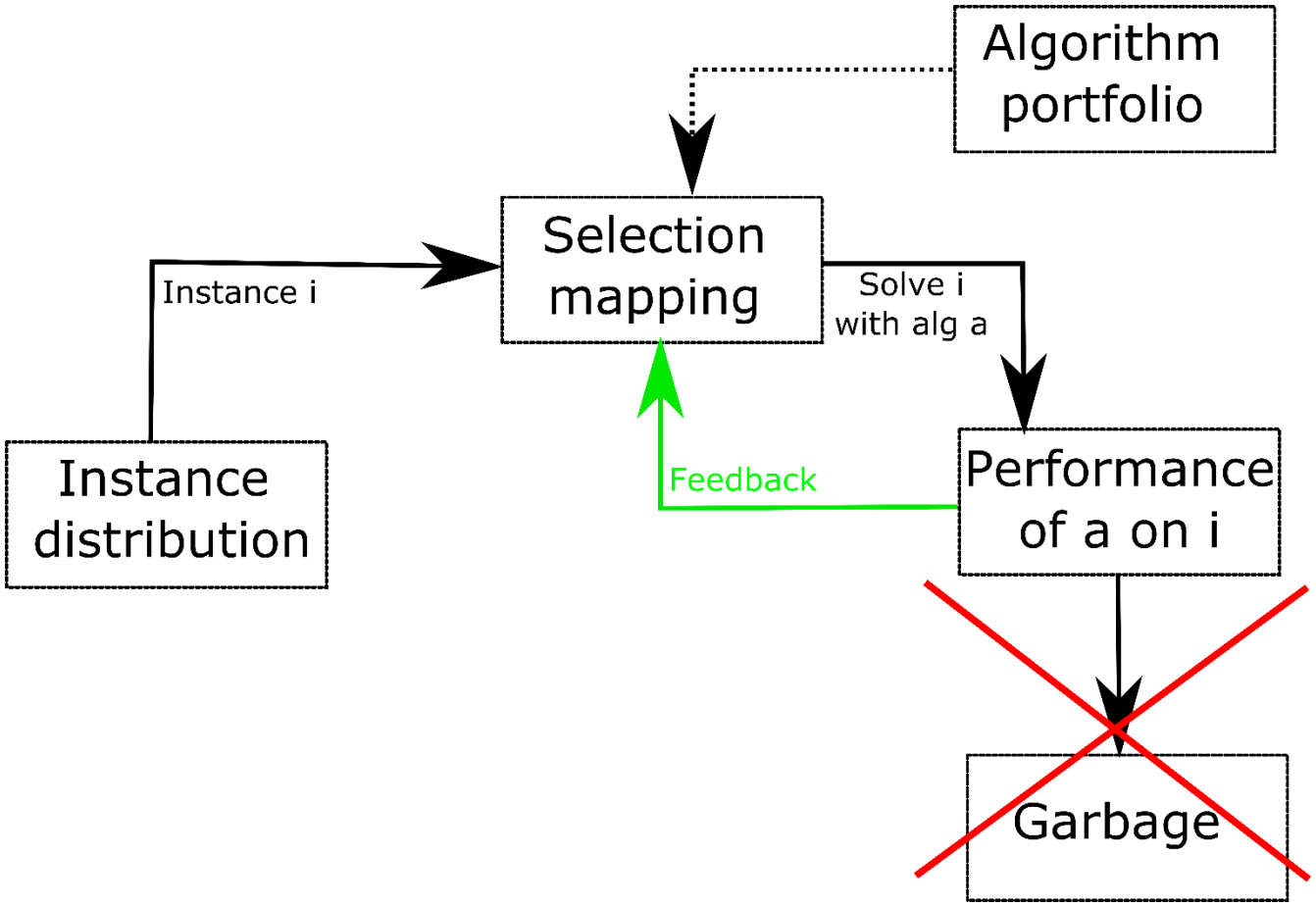
2 h

+

10 m

Can we use this data to further improve the selection mapping?

Automatic Online Algorithm Selection



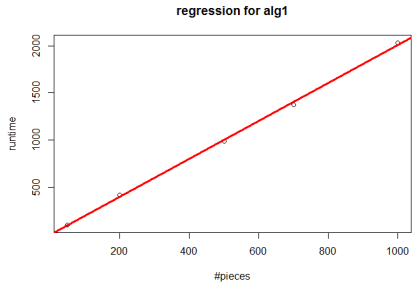
RQ 1: Can online data be used to further improve the selection mapping?

Automatic Online Algorithm Selection (metaphor)

Our puzzlers are used to solve new puzzles for which they are predicted to be best, generating new performance data

+

2 h



Can we use this data to further improve the selection mapping for **regression-based methods**?

Yes we can! Update the regression model corresponding to the selected algorithm

Automatic Online Algorithm Selection (metaphor)

Our puzzlers are used to solve new puzzles for which they are predicted to be best, generating new performance data


+


2 h


+


???

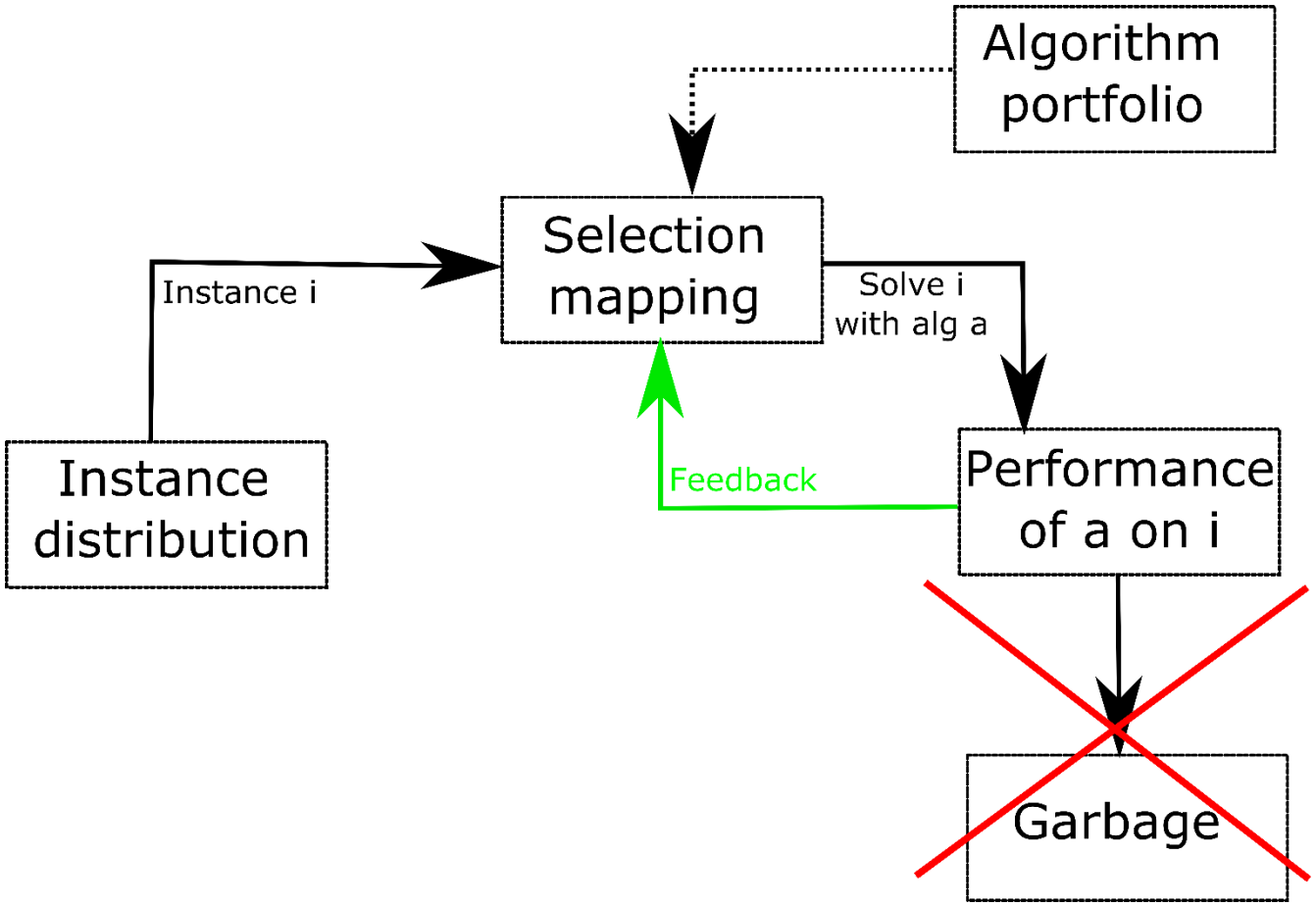

+


???

Can we use this data to further improve the selection mapping for **direct classification methods**?

No we cannot! We do not know the correct classification

Automatic Online Algorithm Selection



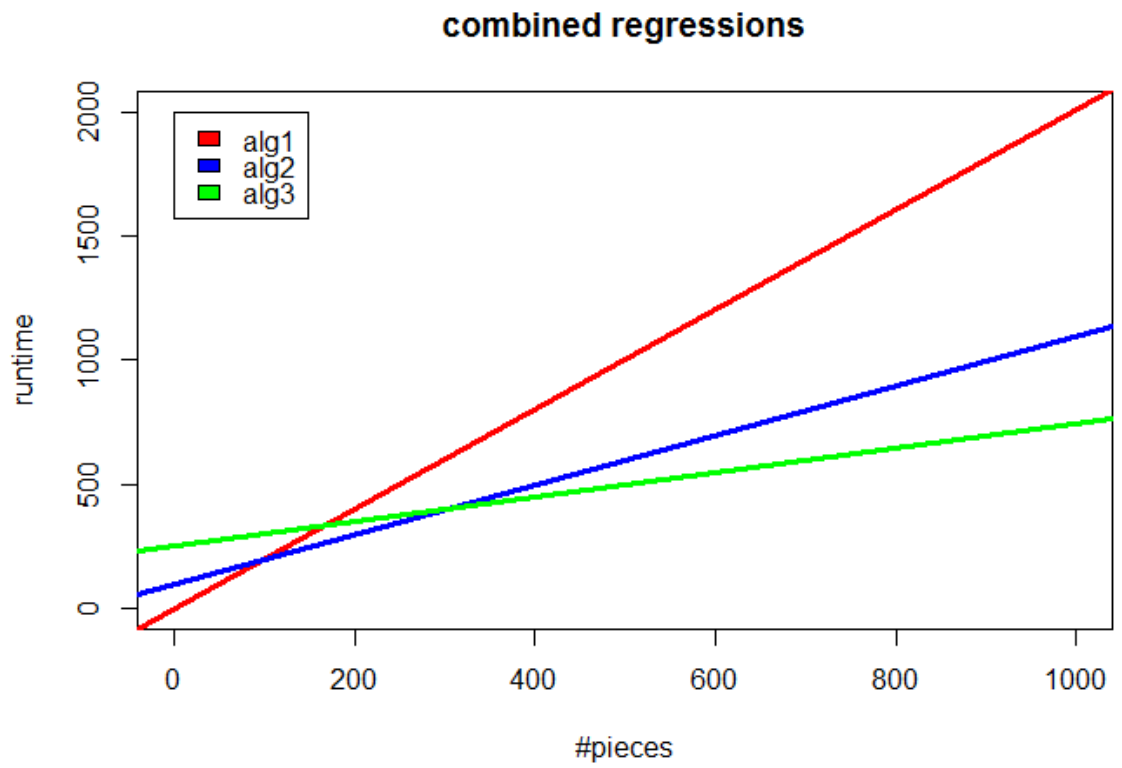
RQ 1: Can online data be used to further improve the selection mapping?

Yes, but only for regression-based methods!

Potential Methodological Issue

- Regression models are updated based on biased samples

  $\Rightarrow$Only new samples for instances on which they are predicted best



Characteristics of the Regression Problem

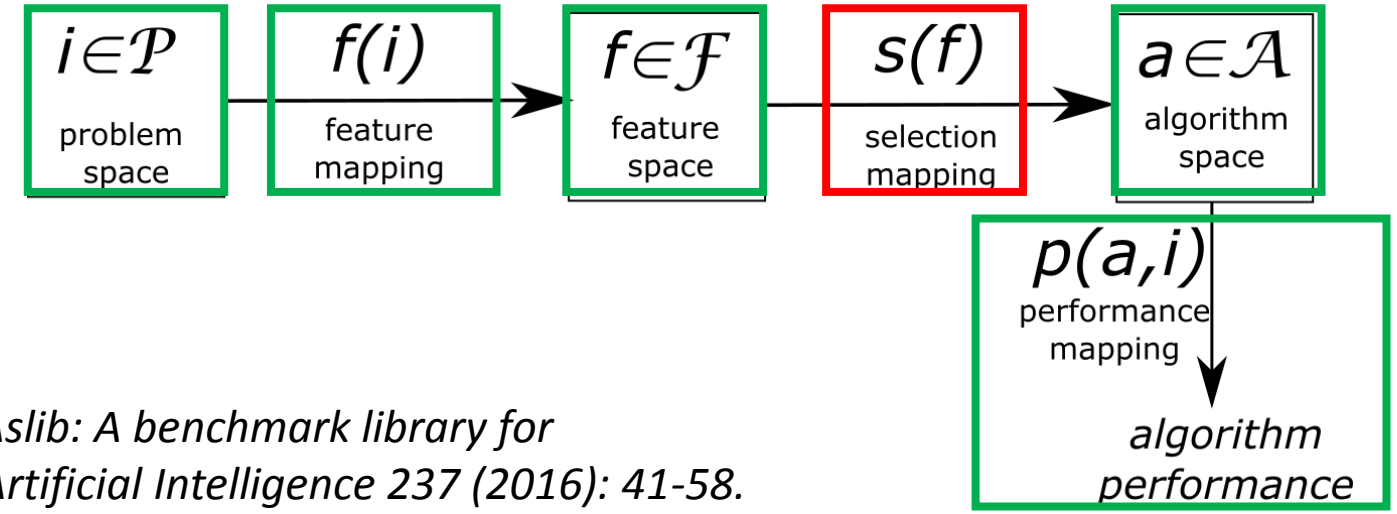
- Problem features
  - Continuous
  - Discrete
  - Boolean
  - …
- Performance is not linear in function of the features
- Regression random forest appears to perform quite well
  - Let me know if you know of other good applicable regression methods
    - Preferably with variance estimates

Automatic Online Algorithm Selection Methodology – Empirical Verification

- All empirical results that follow are the result of a joint work
- Collaborators:
  - Bernd Bischl (Department of Statistics, LMU Munich)
  - Lars Kotthoff (University of British Columbia, Department of Computer Science)

Automatic Online Algorithm Selection Methodology – Empirical Verification

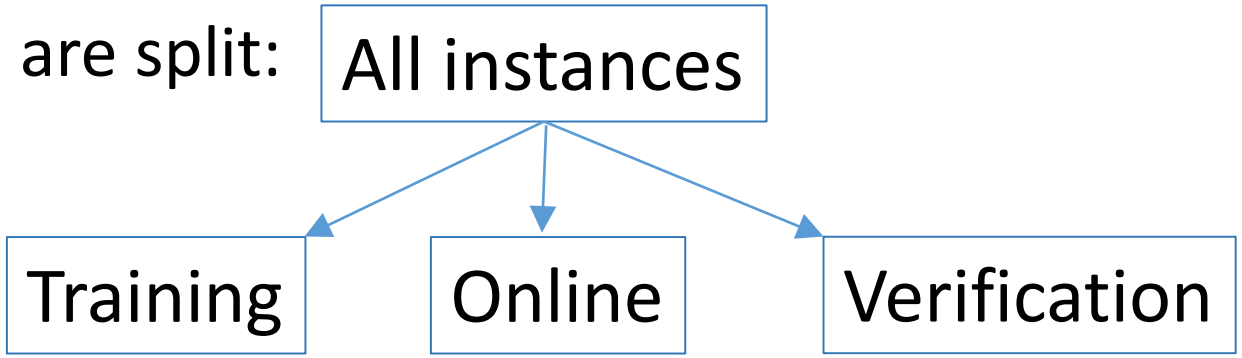
- Does the theory also work in practice?
  - Verify with a simulation study
- ASLIB benchmark instances
  - 15 scenarios available
  - For each scenario: performance of multiple algorithms on the same benchmark instances + feature values for each instance



-Bischl, Bernd, et al. "Aslib: A benchmark library for algorithm selection." Artificial Intelligence 237 (2016): 41-58.

Simulation Study - Experimental Setup

- Instances are split:



- Regression models: **Regression random forest**
- Performance is measured as **PAR10**
- Performance is normalised
  - Virtual Best Solver (VBS) = 0
  - Single Best Solver = 1

Virtual Best Puzzler/ Single Best Puzzler Example

				Virtual Best Puzzler
	2 min	10 min	45 min	2 min
	120 min	30 min	60 min	30 min
	110 min	65 min	30 min	30 min
Avg	77 min	35 min	45 min	21 min

Single Best Puzzler

Simulation Study - Experimental Setup

- Presented here: QBF-2011 results
  - Quantified Boolean Formula competition of 2011
  - 5 algorithms
  - 1368 instances
  - 46 features

- VBS average PAR10: 8337
- Single best average PAR10: 15330

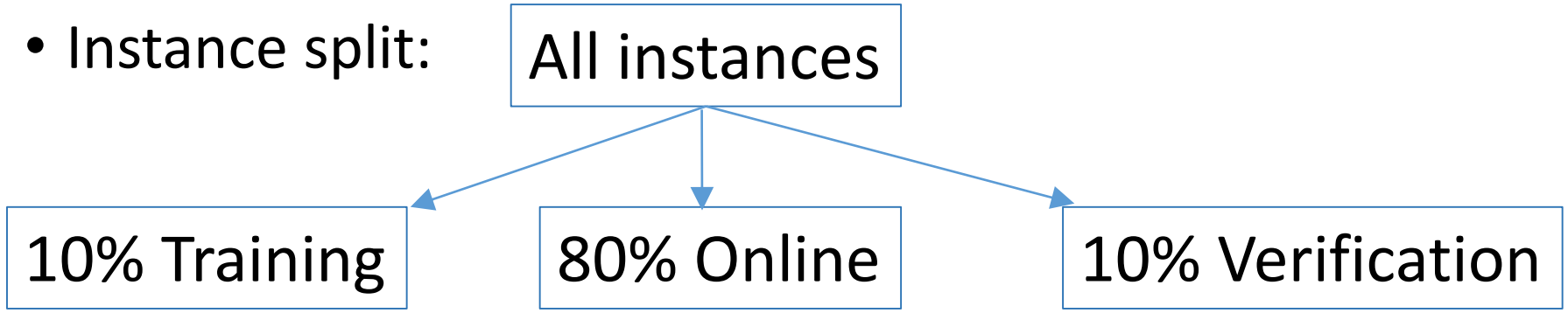
$\Rightarrow$Potential for algorithm selection: 15330-8337 = **6993/instance**

Simulation Study - Experimental Setup

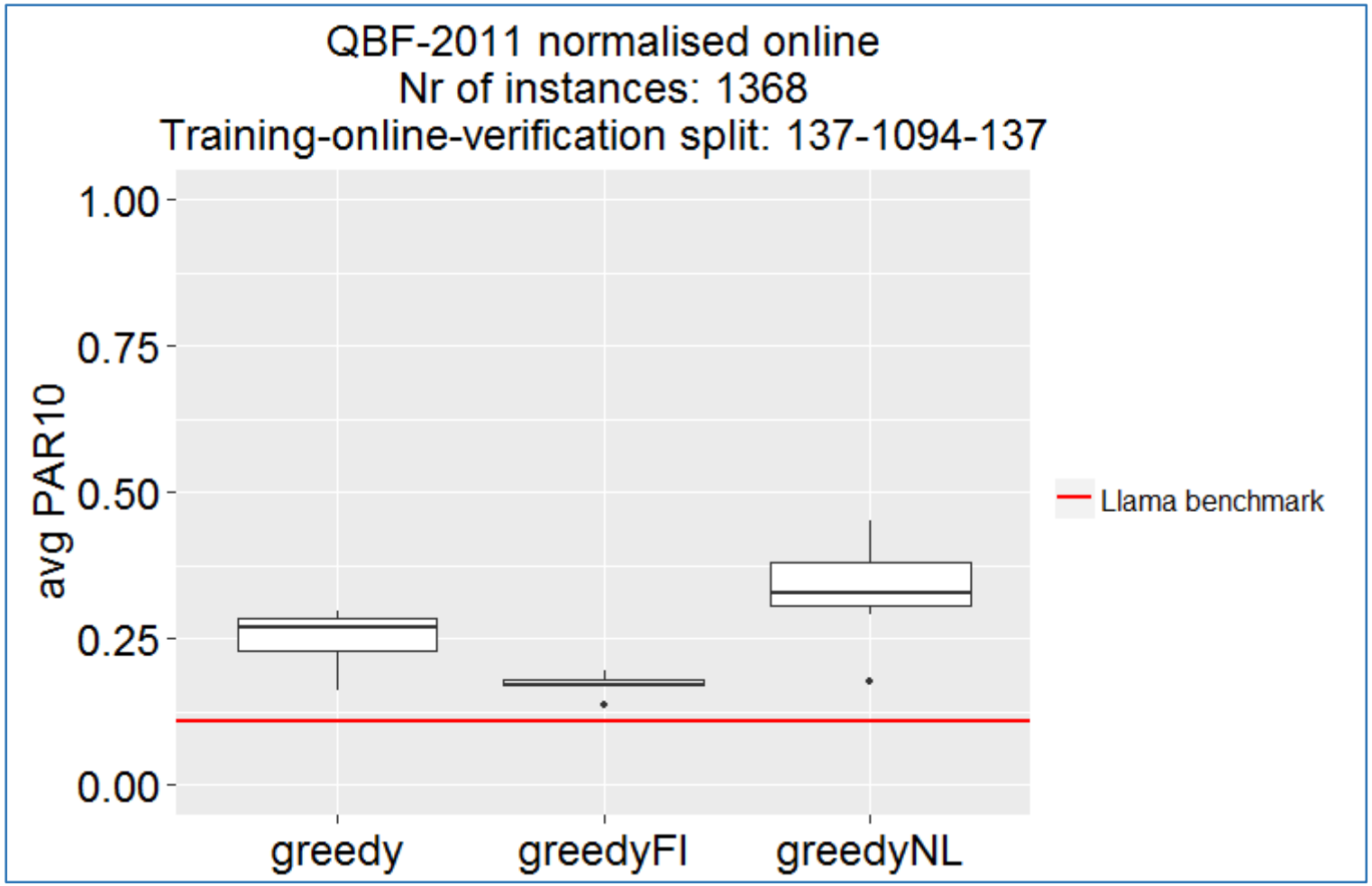
- A regression model is retrained when ≥10 new datapoints are available
  - Retraining is time-consuming
- Performance predictions are batched per 10
  - Predicting is time-consuming
- Each experiment is repeated 10 times
  - Sizeable variance on the simulations

Automatic Online Algorithm Selection Methodology – Empirical Verification

- Selection strategies
  - Greedy
    - Select predicted best and add its performance to the corresponding model
  - Greedy full info (artificial)
    - Like greedy but performance of all other algorithms is also observed
  - Greedy no learning
    - Like greedy but models are never retrained
- Instance split:



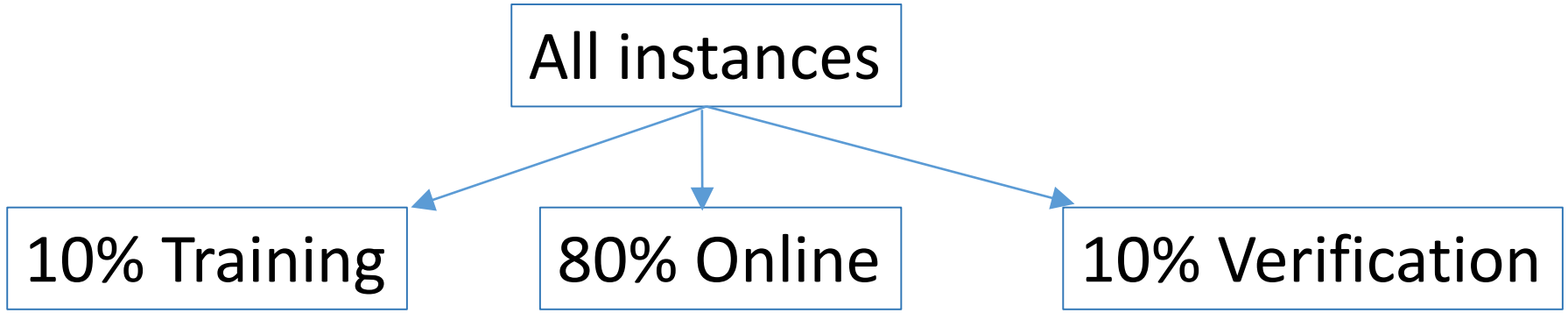
Automatic Online Algorithm Selection Methodology – Empirical Verification



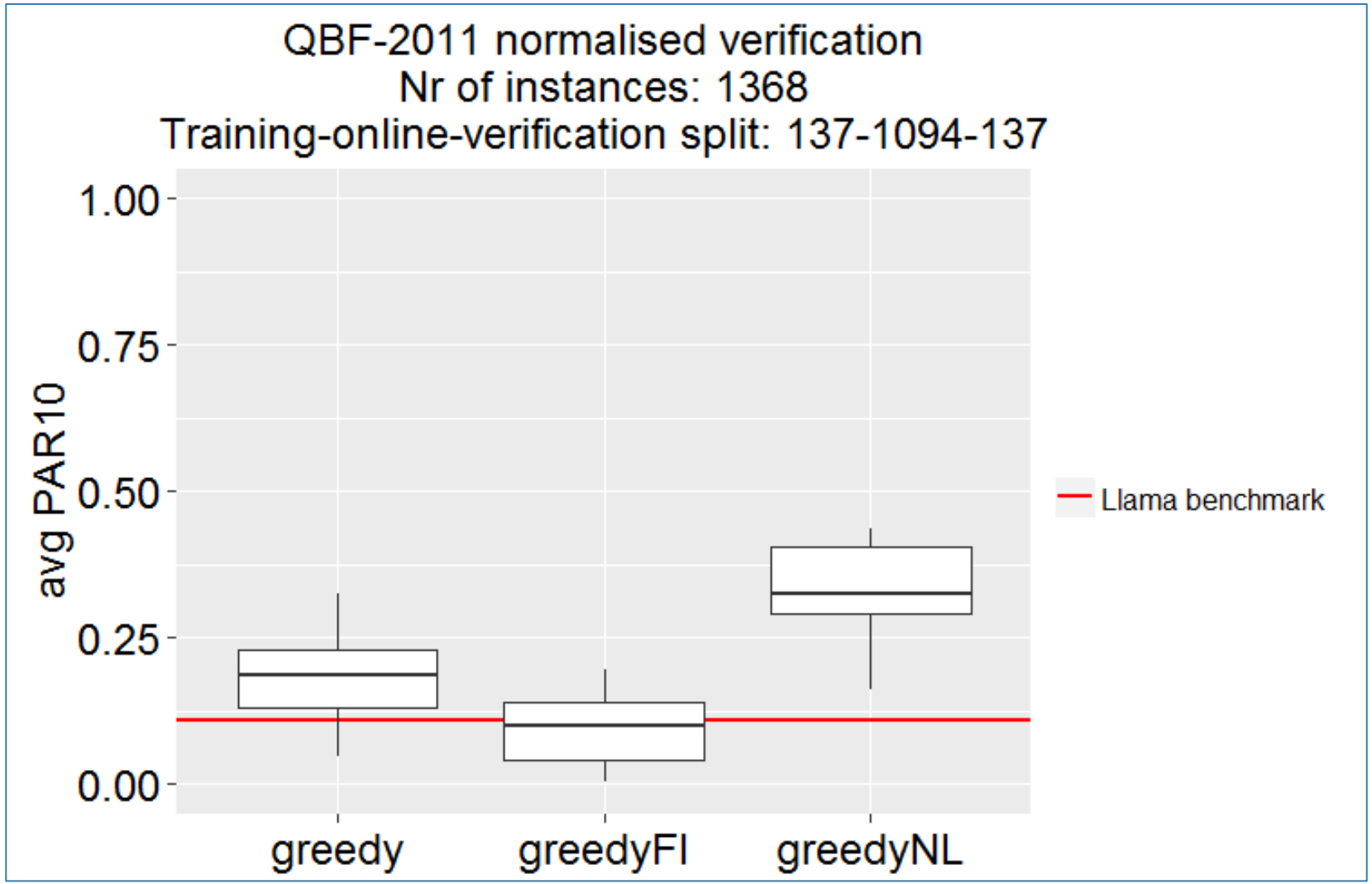
RQ 1: Can online data be used to further improve the selection mapping? Yes!

Automatic Online Algorithm Selection Methodology – Empirical Verification

- **How much has the selection mapping quality improved during the online phase?**
  - Consider for each strategy its models obtained after the online phase
  - Keep the models fixed (no more learning)
  - Run the models on a set of verification instances



How much has the selection mapping quality improved during the online phase?

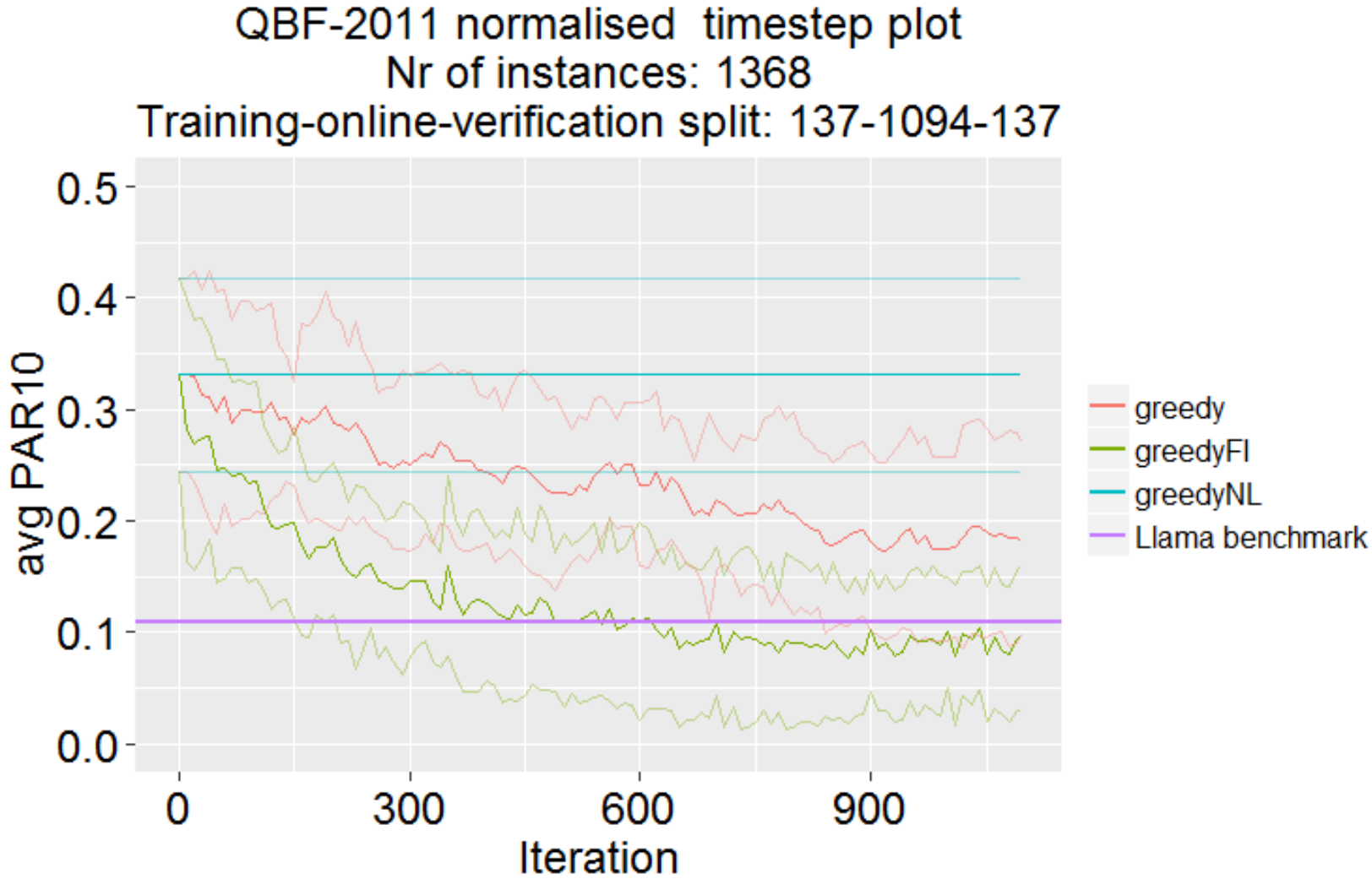


Automatic Online Algorithm Selection Methodology – Empirical Verification

- **How does the selection mapping quality improve over time?**
  - Every 10 timesteps: extract the current models of all selection strategies
  - Keep the models fixed (no more learning)
  - Run the models on a set of verification instances

  $\Rightarrow$Model quality in function of amount of online instances handled

How does the selection mapping quality improve over time?



Intermediary conclusions

- Online data is generated as a natural by-product of algorithm selection
- An automatic online algorithm selection methodology to process this data was introduced
  - Applicable only to regression-based methods
- It was shown empirically that processing online data can improve the selection model
  - Retraining the model from scratch, incorporating the new data
    - Future work: use updateable regression models

What To Expect

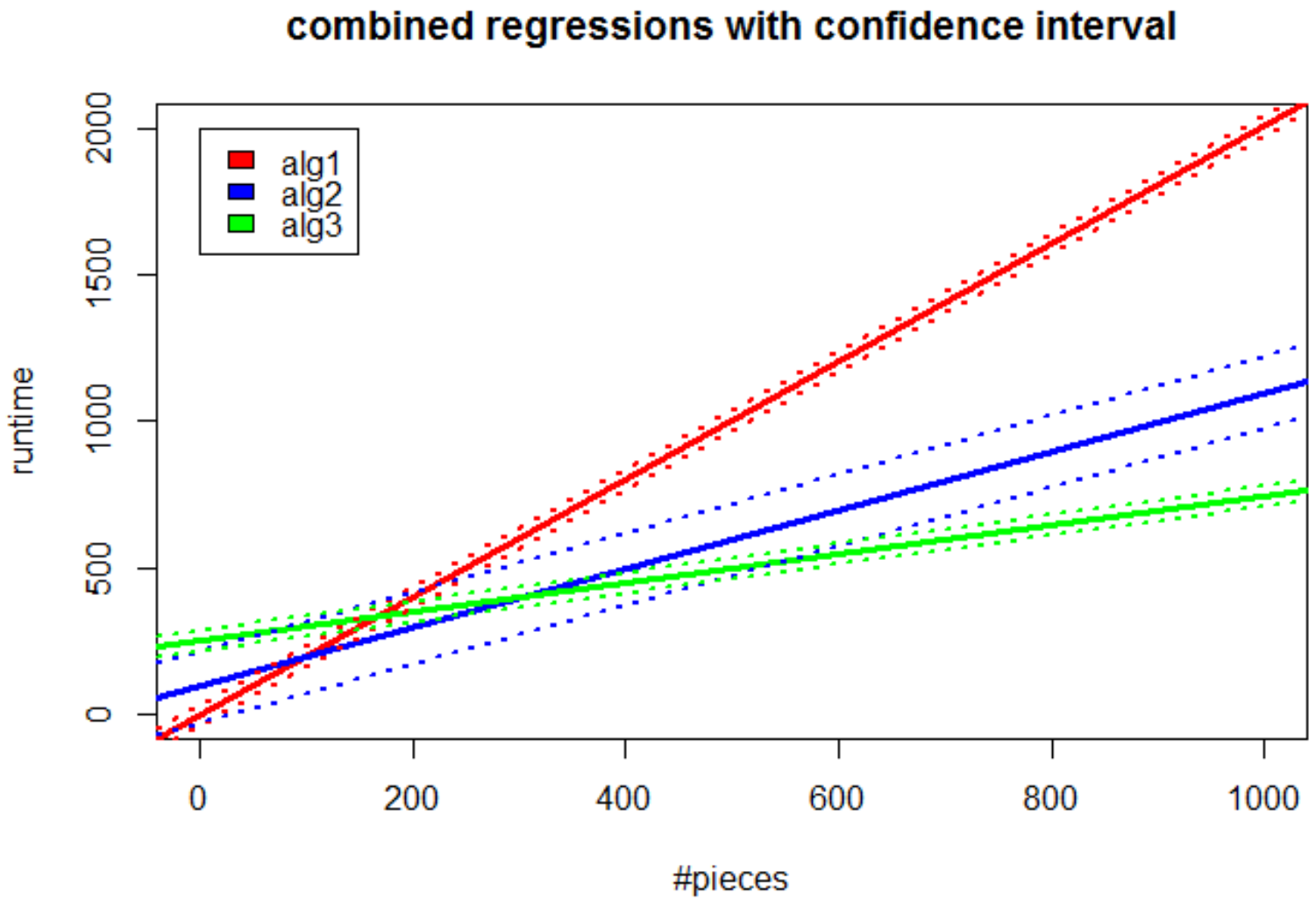
- Automatic Algorithm Selection
- Automatic Online Algorithm Selection
- **Exploration vs. Exploitation and Multi-armed Bandits**
- Exploration vs. Exploitation in Automatic Online Algorithm Selection

Exploration vs. Exploitation Trade-off

- Selecting an algorithm influences two things
  1. Short term: The performance on the current instance
  2. Long term: The model-quality of the selected algorithm (influencing future predictions)
- Short and long term goals can be opposite
  1. Short term: Select predicted best
  2. Long term: Select algorithm with large variance on prediction

$\Rightarrow$**Exploration vs. exploitation trade-off**

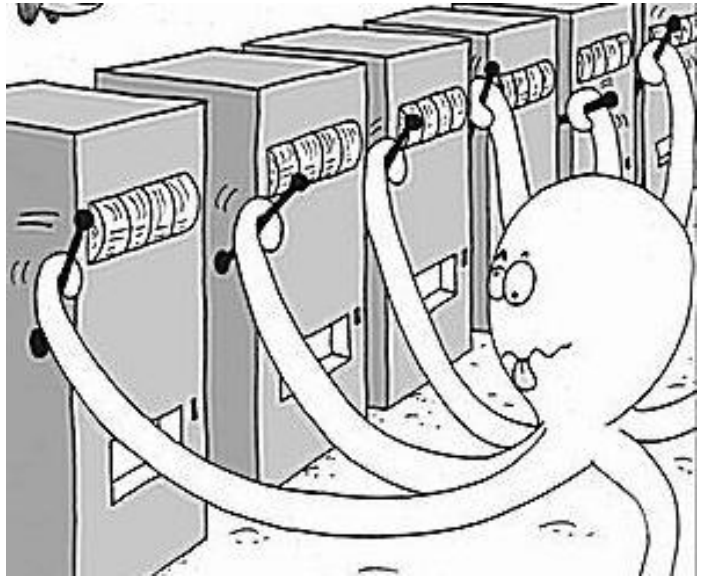
Exploration vs. Exploitation Trade-off (metaphor)



New puzzle
400 pieces
Alg3?
Why not try Alg2!

Multi-armed Bandits

- N available gambling machines
  - With unknown but fixed reward distribution
- A gambler pulls machines and receives rewards
  - A sample from the corresponding distribution
- Gambler’s goal: maximise expected profit
  - Over finite time-horizon
  - Over infinite time-horizon



Multi-armed Bandits

GOOD: 5 BAD: 3

GOOD: 1 BAD: 2

GOOD: 2 BAD: 2

Which arm to pull next?

Best for learning which is best? (exploration)

Best for largest expected reward? (exploitation)

Multi-armed Bandits

GOOD: 501 BAD: 500

GOOD: 1 BAD: 2

GOOD: 2 BAD: 2

Which arm to pull next?

Best for learning which is best? (exploration)

Best for largest expected reward? (exploitation)

Multi-armed Bandits

- Investigated by computer scientists and mathematicians
- Optimal solutions exist
  - Asymptotical convergence to the best arm
  - Bound on the total regret incurred
    - Regret is incurred every time a non-best arm is pulled

Auer, Peter, Nicolo Cesa-Bianchi, and Paul Fischer. "Finite-time analysis of the multiarmed bandit problem." Machine learning 47.2-3 (2002): 235-256.

Multi-armed Bandits: Solution Methods

- **Greedy**: Always select best
- **ε-Greedy**: Random with probability ε, greedy otherwise
  - ε can be decreasing over time for a more optimal solution
- **UCB-** $\boldsymbol{\lambda}$ (Upper Confidence Bound): Select algorithm with highest value for: $mean+variance*\lambda$
- **Thompson Sampling**: Pick each arm with probability equal to its probability of being best

Website Design as a Multi-Armed Bandit Problem

- Goal
  - Create attractive website
- Method
  - Create alternative web pages and observe visitor behaviour on each of them
- As a multi-armed bandit problem
  - A web page alternative = a bandit
  - Pulling a handle = showing an alternative to a visitor
  - Reward = Does the visitor do what you want him to? (Stay, buy, press ok…)
- Why relevant?
  - The business model of a lot of IT companies is advertisement based

Contextual Multi-Armed Bandit Problem

- Context is shown every time an arm is pulled
  - Ex: the gender and age of a visitor on a webpage
- Depending on context a different arm will be pulled

- Much harder than the standard Multi-Armed Bandit problem
- Solution strategy: regression model for each reward distribution
  - LinUCB algorithm: assumes linear rewards

Chu, Wei, et al. "Contextual Bandits with Linear Payoff Functions." AISTATS. Vol. 15. 2011.

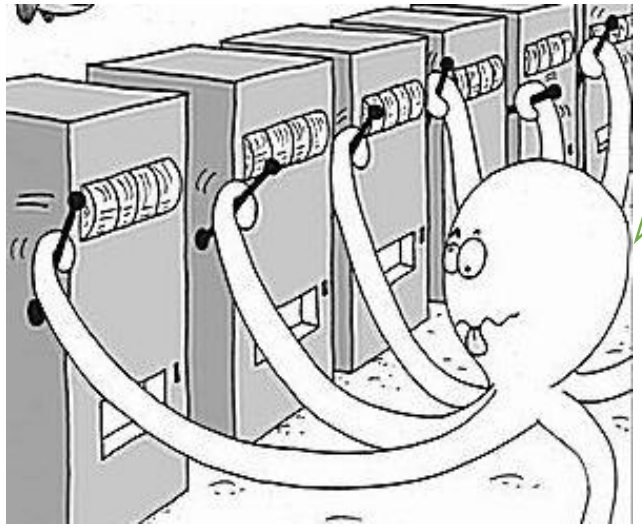
What To Expect

- Automatic Algorithm Selection
- Automatic Online Algorithm Selection
- Exploration vs. Exploitation and Multi-armed Bandits
- **Exploration vs. Exploitation in Automatic Online Algorithm Selection**

Automatic Online Algorithm Selection as a Contextual Multi-armed Bandit

It is better to be lucky. But I would rather be exact. Then when luck comes you are ready.

Arms = Algorithms
Context = Features
Pulling an arm = Selecting an algorithm for an instance



Work Related To Automatic Online Algorithm Selection

- *Cicirello, Vincent A., and Stephen F. Smith. "The max k-armed bandit: A new model of exploration applied to search heuristic selection." AAAI. 2005.*
- Solve one instance as good as possible within a budget of n tries while having access to a set of stochastic algorithms

Arms = Algorithms
Context = None
Pulling an arm = Running an algorithm on the instance

Work Related To Automatic Online Algorithm Selection

- *Gagliolo, Matteo, and Jürgen Schmidhuber. "Algorithm selection as a bandit problem with unbounded losses." International Conference on Learning and Intelligent Optimization. Springer Berlin Heidelberg, 2010.*
  - Goal: Learn the best parallel portfolio time allocation-function

Arms = Time-allocators
Context = None
Pulling an arm = Selecting a time-allocator for an instance

Exploration vs. Exploitation Trade-off

- Selecting an algorithm influences two things
  1. Short term: The performance on the current instance
  2. Long term: The model-quality of the selected algorithm (influencing future predictions)
- Short and long term goals can be opposite
  1. Short term: Select predicted best
  2. Long term: Select algorithm with large variance on prediction

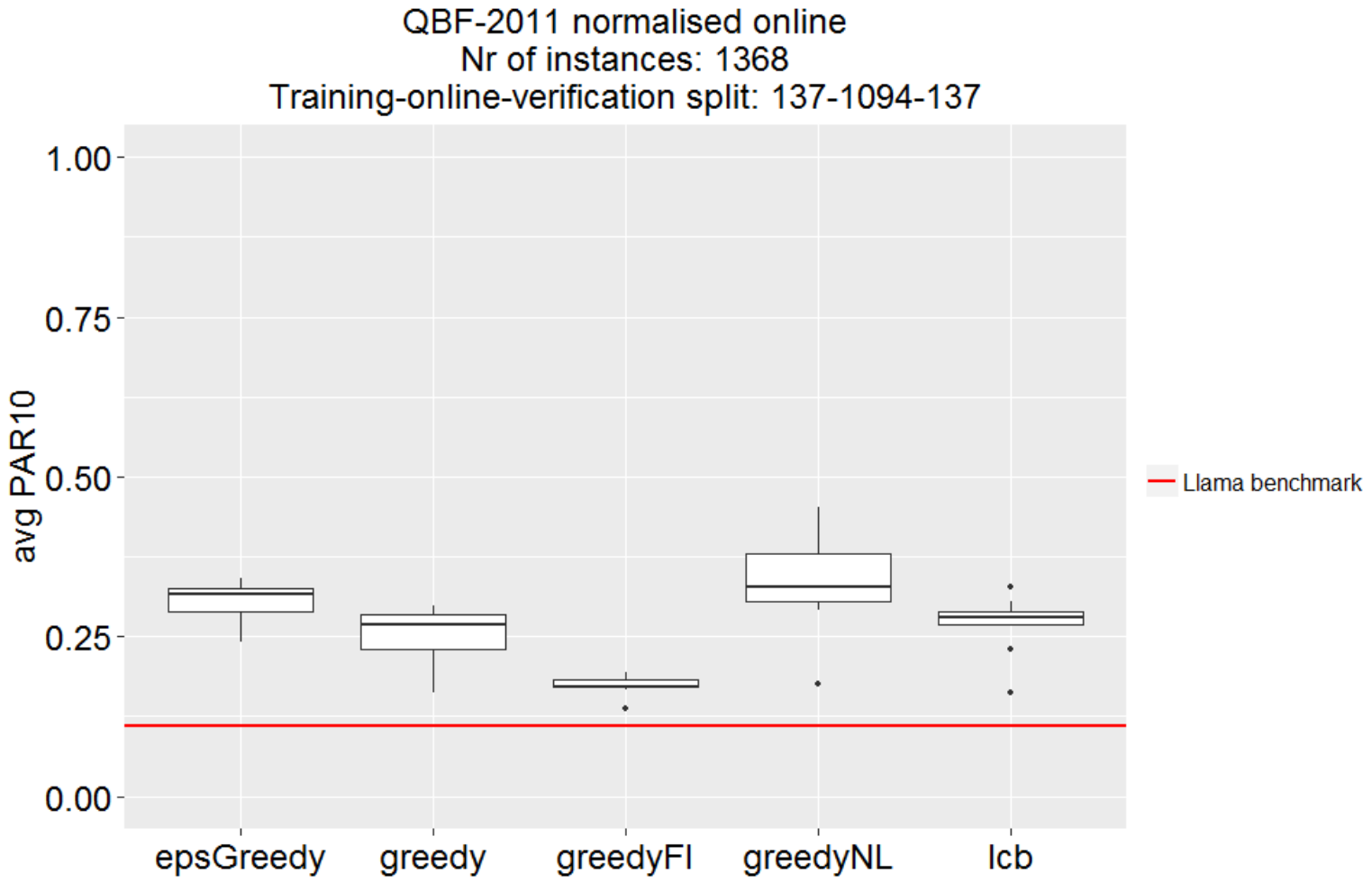
$\Rightarrow$**Exploration vs. exploitation trade-off**

RQ 2: Can the exploration vs. exploitation trade-off be empirically shown to exist?

Exploration vs. Exploitation Trade-off: Empirical Verification

- Selection strategies
  - Greedy
    - Select predicted best and add its performance to the corresponding model
  - Greedy full info
    - Like greedy but performance of all other algorithms is also observed
  - Greedy no learning
    - Like greedy but models are never retrained
  - ε-Greedy
    - Random with probability ε, greedy otherwise
    - ε = 0.05
  - LCB-λ (Lower Confidence Bound)
    - Select algorithm with lowest value for: mean-variance*λ
    - $\lambda = 1$

Exploration vs. Exploitation Trade-off: Empirical Verification

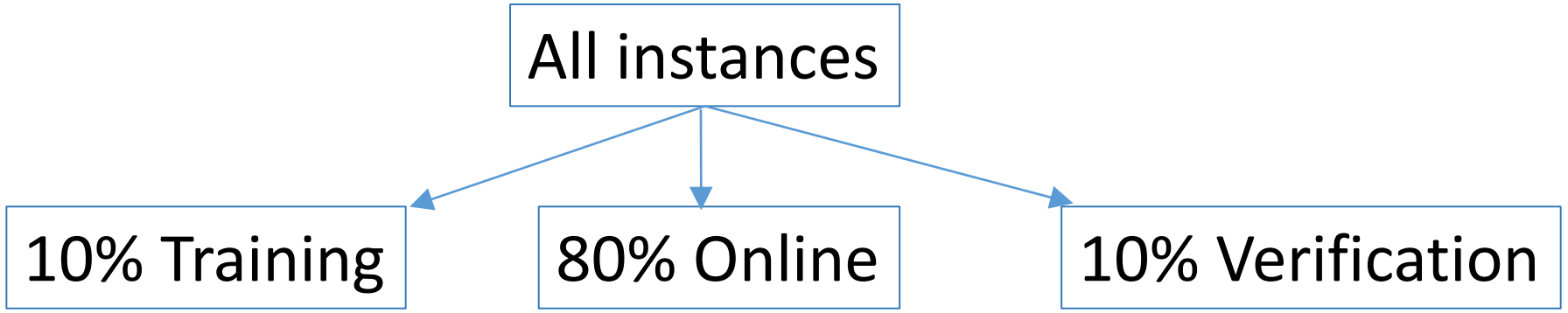


RQ 2: Can the exploration vs. exploitation trade-off be empirically shown to exist? No

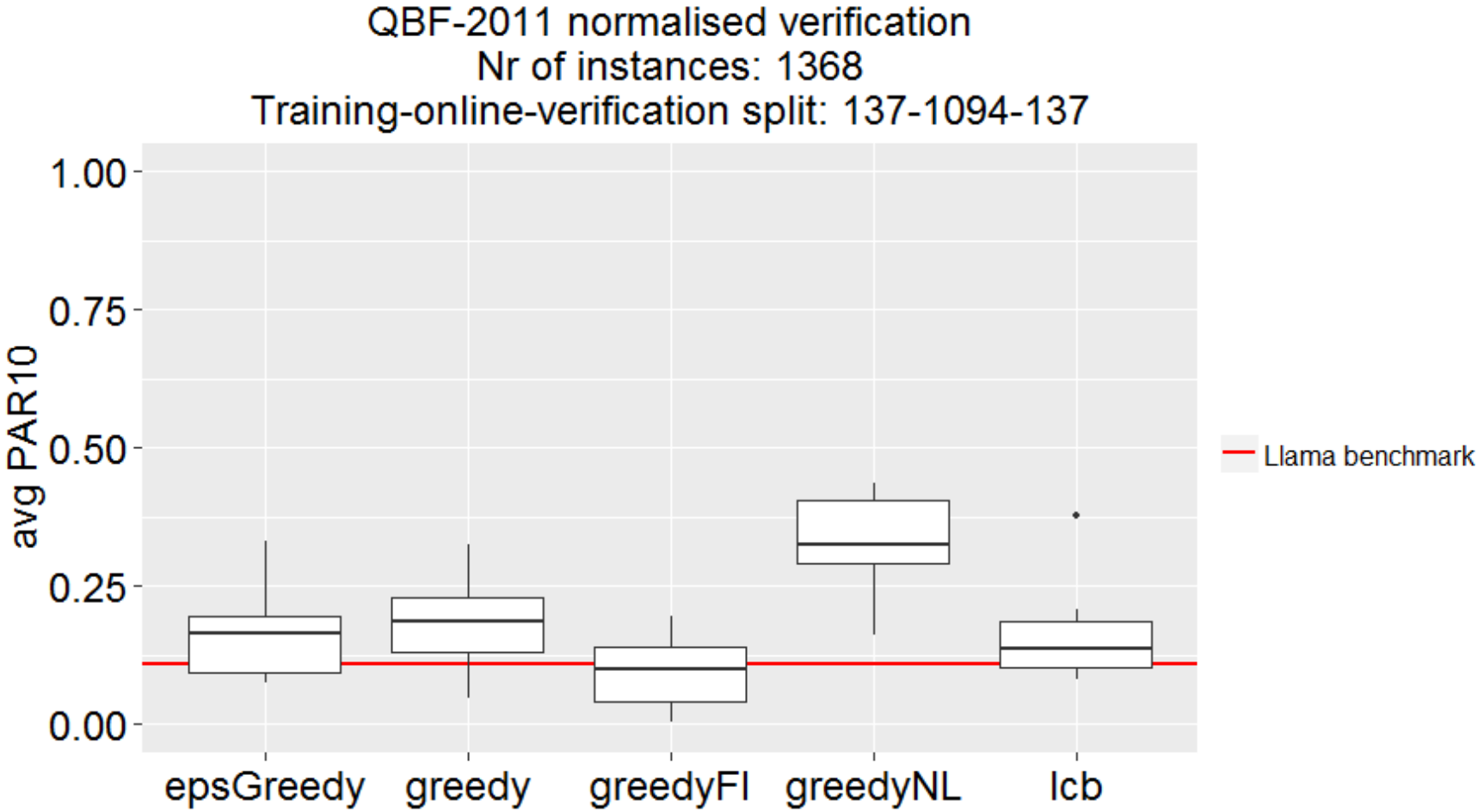
Exploration vs. Exploitation Trade-off: Empirical Verification

- **Have the explicitly exploring ε-Greedy and LCB at least managed to learn a better selection mapping?**
  - Perhaps the performance-loss due to exploring was too big to compensate

  - Consider for each strategy its models obtained after the online phase
  - Keep the models fixed (no more learning)
  - Run the models on a set of verification instances



Have the explicitly exploring ε-Greedy and LCB managed to learn a better selection mapping?



Does explicitly exploring result in better models? Not convincingly

Exploration vs. Exploitation Trade-off: Empirical Verification

- The experiments did not show explicit exploration to be beneficial for solving the exploration vs. exploitation trade-off
- The experiments did not even convincingly show explicit exploration to learn significantly better models
- Unexpected results! Possible explanations:
  1) Amount of online data small compared to amount of offline data
  2) Simple greedy performs implicit exploration
  3) Bad solution methods to the exploration-exploitation trade-off

Exploration vs. Exploitation Trade-off: Empirical Verification

- Unexpected results! Possible explanations:
  1) **Amount of online data small compared to amount of offline data**
  2) Simple greedy performs implicit exploration
  3) Bad solution methods to the exploration-exploitation trade-off

Amount of online data small compared to amount of offline data?

- 137 training instances
  - 5 algorithms each with 137 training datapoints
    - 137 exploitation steps
    - 137*4=548 exploration steps
- 1094 online instances
  - ε-Greedy: 5% exploration
    - ~1039 exploitation steps
    - ~55 exploration steps

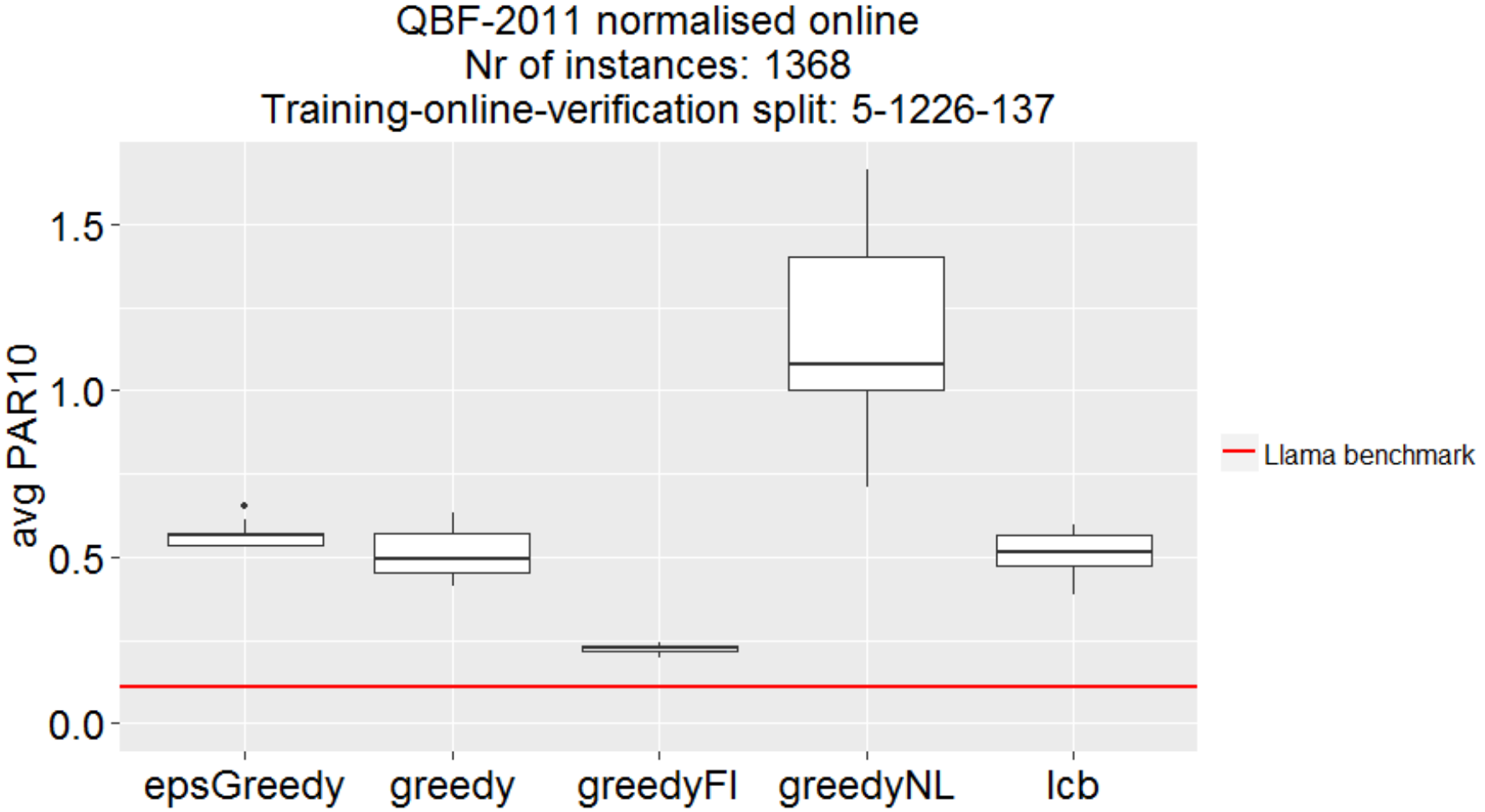
$\Rightarrow$Amount of online exploration is negligible

Amount of online data small compared to amount of offline data?

- 5 training instances
  - 5 algorithms each with 5 training datapoints
    - 5 exploitation steps
    - 5*4=20 exploration steps
- 1226 online instances
  - ε-Greedy: 5% exploration
    - ~1165 exploitation steps
    - ~61 exploration steps

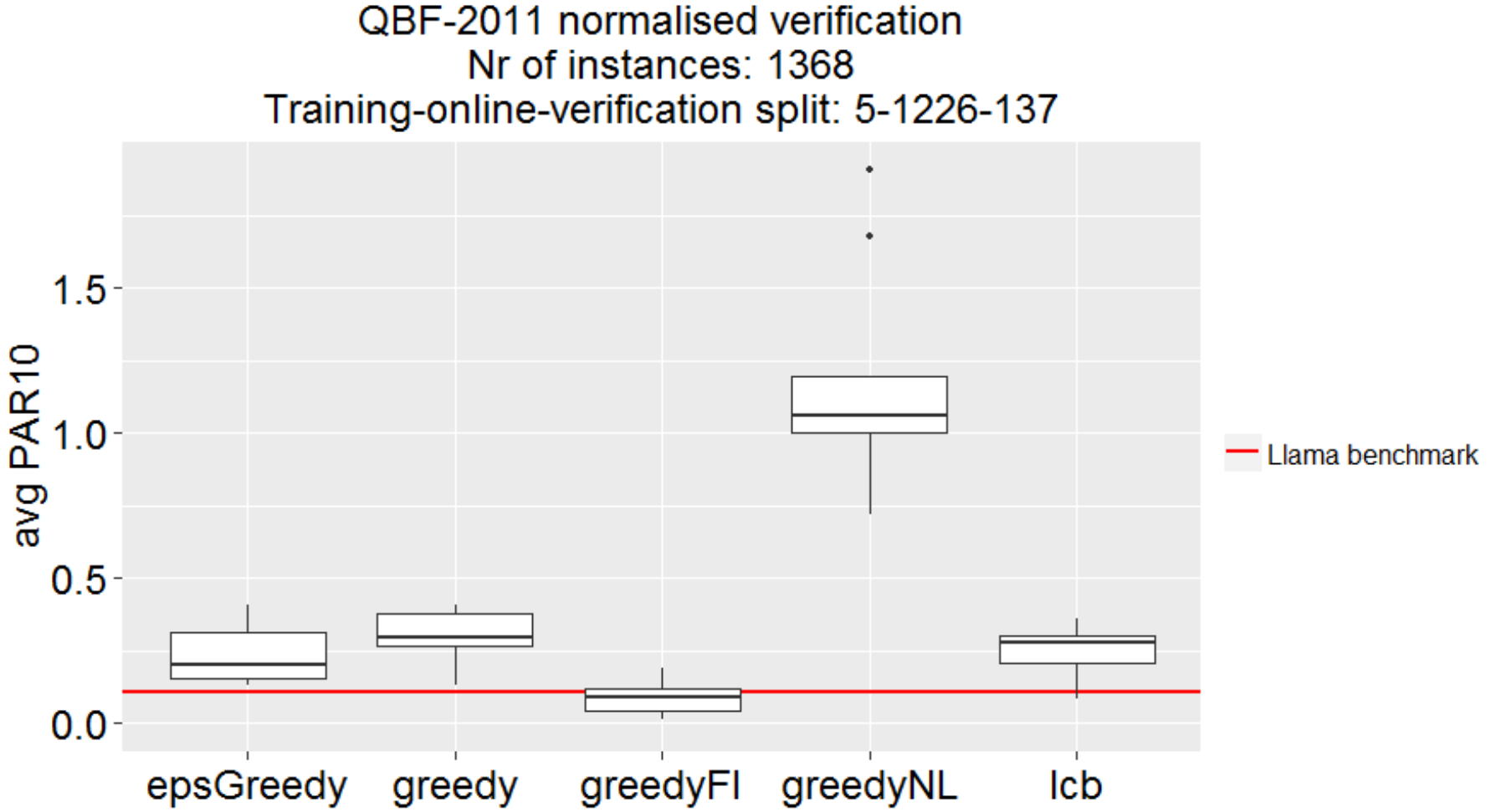
$\Rightarrow$Amount of online exploration is now significant

Exploration vs. Exploitation Trade-off: Empirical Verification



RQ 2: Can the exploration vs. exploitation trade-off be empirically shown to exist? Still not

Have the explicitly exploring ε-Greedy and LCB managed to learn a better selection mapping?



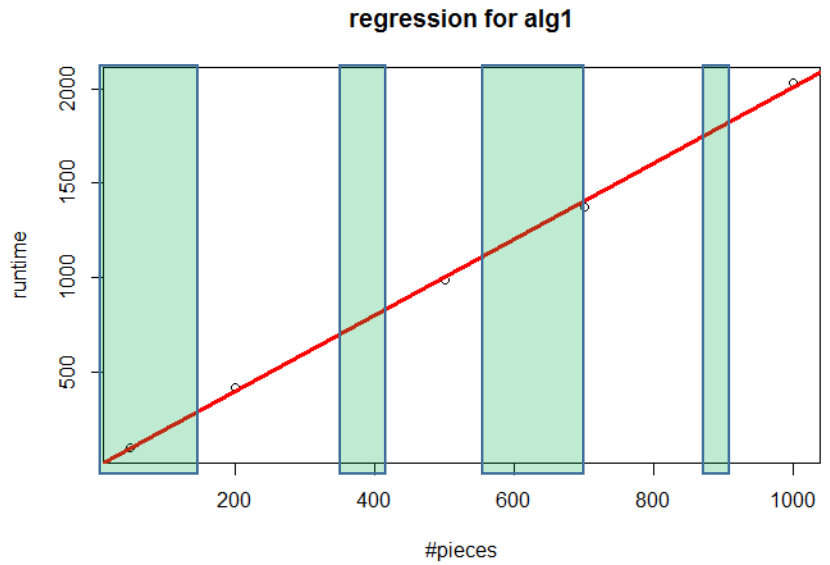
Does explicitly exploring result in better models? Yes (barely)

Exploration vs. Exploitation Trade-off: Empirical Verification

- Unexpected results! Possible explanations:
  1) Amount of online data small compared to amount of offline data
  2) **Simple greedy performs implicit exploration**
  3) Bad solution methods to the exploration-exploitation trade-off

Does simple greedy perform implicit exploration?

- Greedy: An algorithm is only selected to solve an instance when it is predicted best
- But: Each algorithm has a regression model making predictions over the entire instance space
  - New datapoints in the region where an algorithm is predicted to be best also improve predictions in other regions



Investigating this is future work

Exploration vs. Exploitation Trade-off: Empirical Verification

- Unexpected results! Possible explanations:
  1) Amount of online data small compared to amount of offline data
  2) Simple greedy performs implicit exploration
  3) **Bad solution methods to the exploration-exploitation trade-off**

Are the proposed solution methods to the exploration vs. exploitation trade-off bad?

- ε-Greedy is a provably bad strategy
  - But ε-Greedy with decreasing ε is not
- UCB/LCB should be decent
  - Need insight of multi-armed bandit expert to confirm

- There exist solutions to the contextual multi-armed bandit problem with proven guarantees
  - LinUCB: assumes linear rewards
  - Others that don’t assume linearity?

Testing alternative strategies is future work

Exploration vs. Exploitation Trade-off: Empirical Verification

- Unexpected results! Possible explanations:
  1) Amount of online data small compared to amount of offline data
     - Partial explanation
  2) Simple greedy performs implicit exploration
     - Future work
  3) Bad solution methods to the exploration-exploitation trade-off
     - Future work

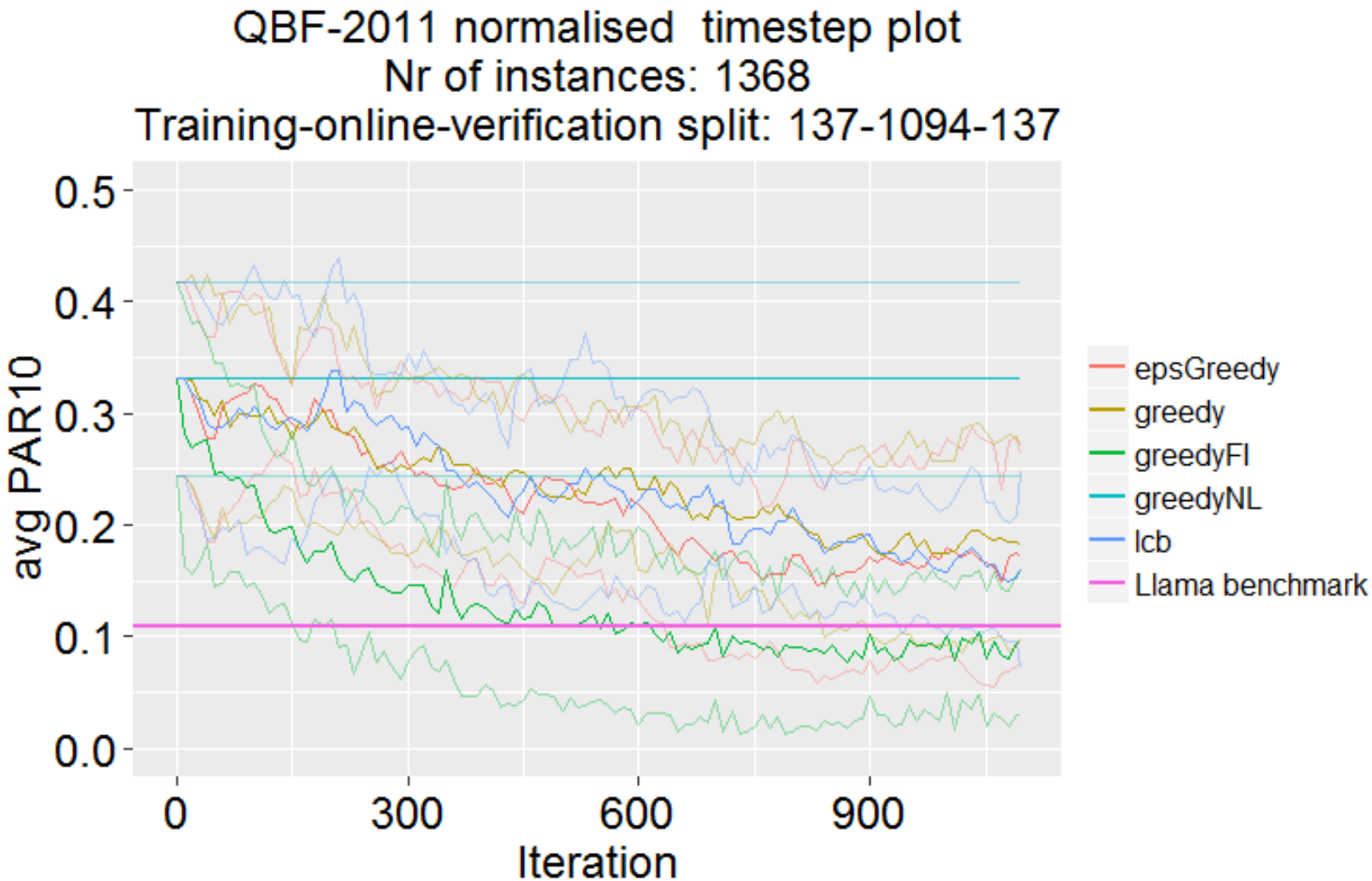
- It would be interesting to see how model quality evolves over time

Exploration vs. Exploitation Trade-off: Empirical Verification

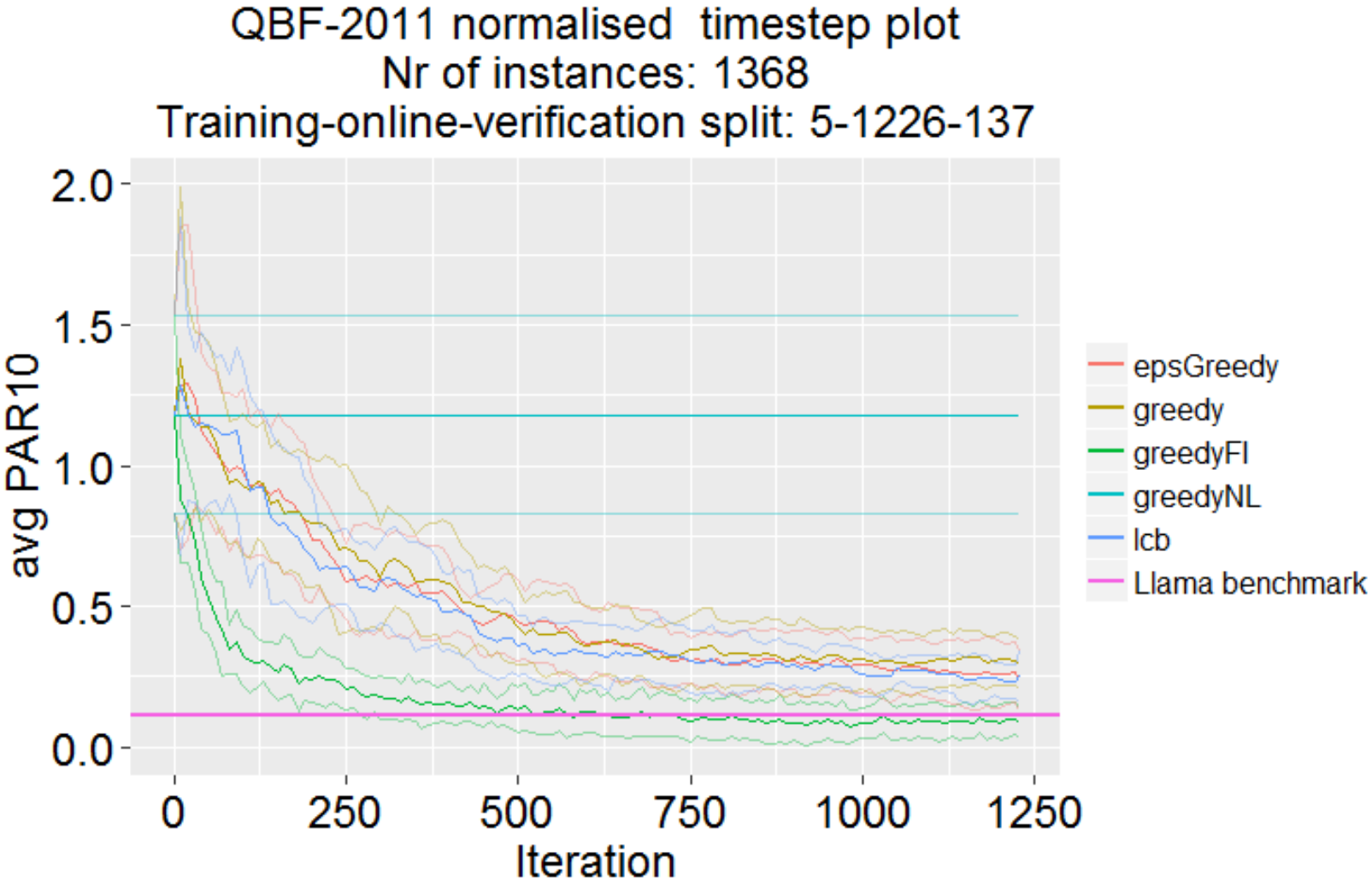
- **How does the selection mapping quality improve over time?**
  - Every 10 timesteps: extract the current models of all selection strategies
  - Keep the models fixed (no more learning)
  - Run the models on a set of verification instances

  $\Rightarrow$Model quality in function of amount of online instances handled

How does the selection mapping quality improve over time? (10% training data)



How does the selection mapping quality improve over time? (5 training instances)

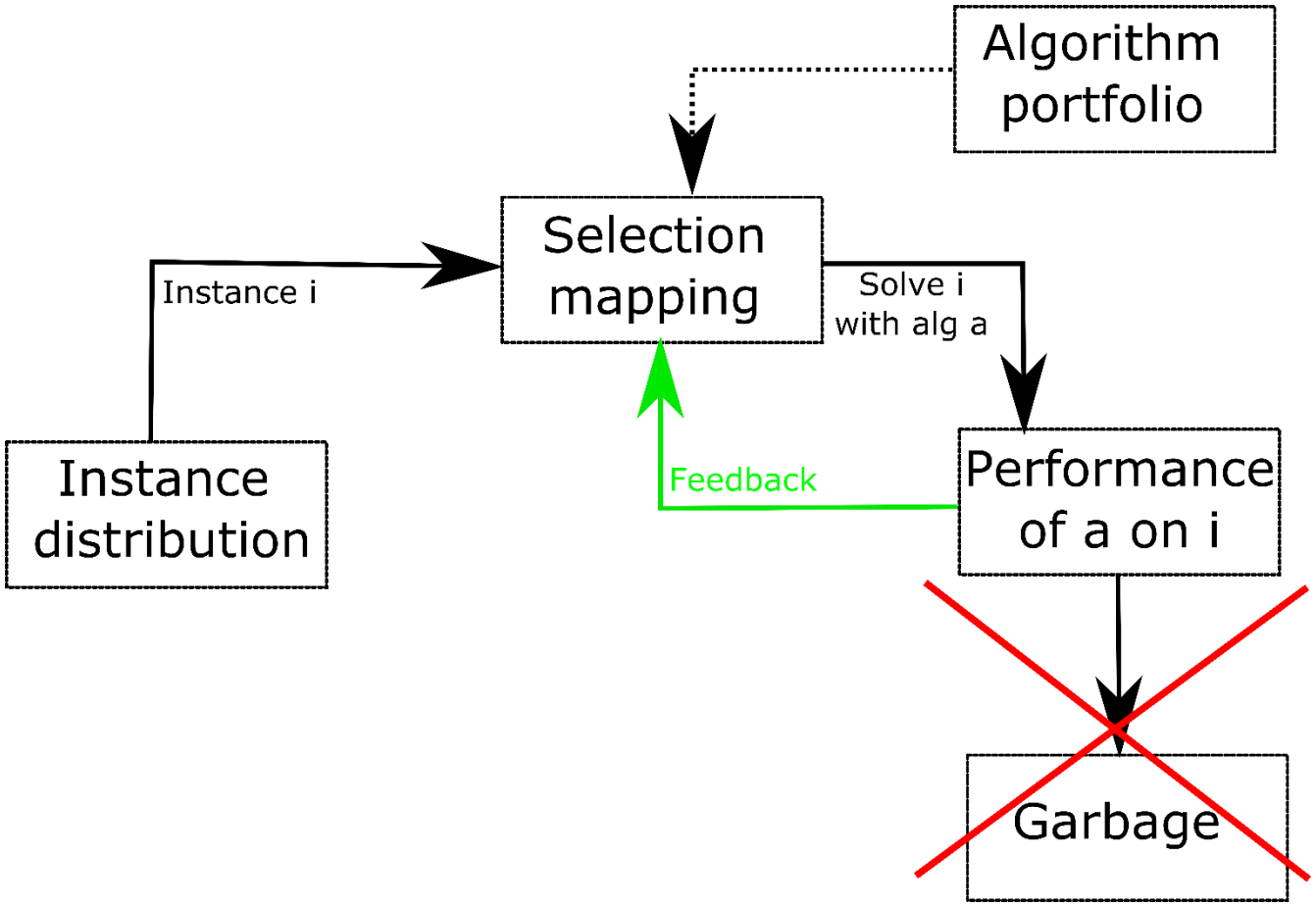


How does the selection mapping quality improve over time?

- Value of new data decreases over time
- Very little data is needed to create a selection model that outperforms the single best solver (greedyFI)

- Greedy, ε-Greedy and LCB seem equivalent
  - More influence of variance than of selection strategy

Conclusions



RQ 1: Can online data be used to further improve the selection mapping?

RQ 2: Can the exploration vs. exploitation trade-off be empirically shown to exist?

Conclusions

- Online data is generated as a natural by-product of algorithm selection
- An automatic online algorithm selection methodology to process this data was introduced
  - Applicable only to regression-based methods
- It was shown empirically that processing online data can improve the selection model
- Automatic online algorithm selection can be modelled as a contextual multi-armed bandit problem
- The exploration vs. exploitation trade-off has not convincingly been empirically shown to exist

Take-aways

- Automatic algorithm selection can be useful even with a small amount of training data
- The straightforward extension to automatic online algorithm selection using the simple greedy selection strategy works well
  - Caveat: needs further research

Hans.Degroote@kuleuven.be

If you have data of ≥2 algorithms for a problem, feel free to mail me. I can check the potential for algorithm selection