Online Algorithm Selection

Hans Degroote[1], Bernd Bischl[2], Lars Kotthoff[3] and Patrick De Causmaecker[1]

[1]KU Leuven, Department of Computer Science - CODeS & imec-ITEC

[2]Department of Statistics, LMU Munich

[3]University of Wyoming, Department of Computer Science

Overview

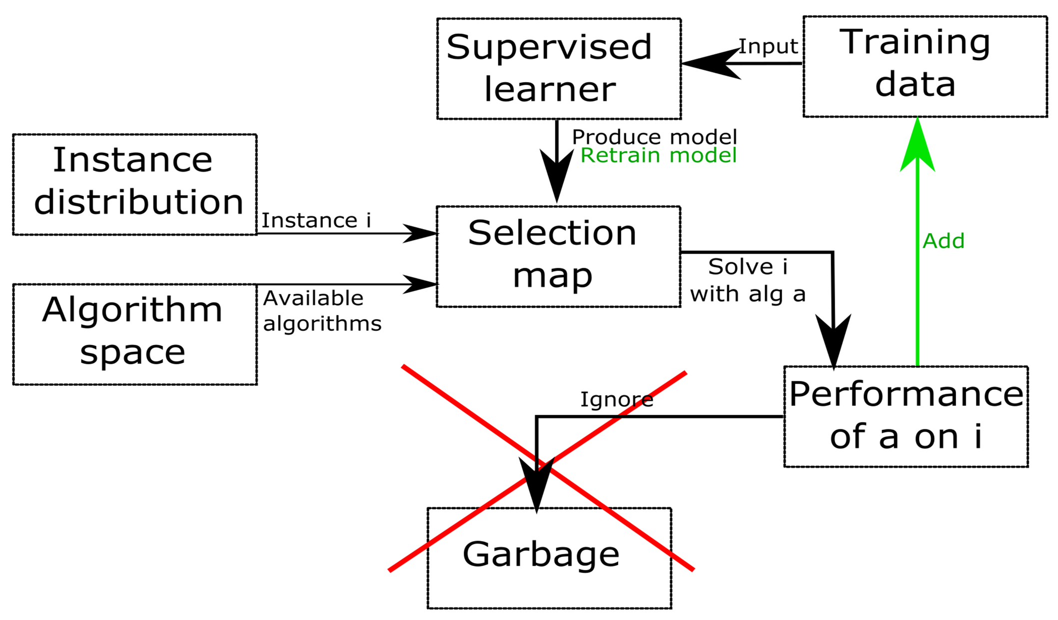
Algorithm selection attempts to select for each problem instance the best algorithm
Classic approach: supervised learning
Observation: when selecting an algorithm for a new instance, its performance becomes known
Idea: use this data to improve the model
Contributions

- Online algorithm selection can be modelled as a contextual bandit
- A methodology for online algorithm selection
- Empirical verification of methodology on ASLIB
  ⇒Processing online data results in better models, but a simple greedy approach outperforms exploring alternatives

Online Algorithm Selection

Motivation:

- **Selection mapping usually not optimal** after training (biased or incomplete training data)
- **Free feedback data** is generated while performing algorithm selection



Goal: use the free feedback performance data to keep improving the selection mapping
Side advantage: over time the true instance distribution is approached better and better
Hypothesis: the proposed methodology can also handle a changing instance distribution

As a Contextual Bandit

Online algorithm selection		**Contextual bandit**
Observing feature values	=	Seeing a context
Selecting an algorithm		Pulling an arm
Observing performance		Obtaining a reward

⇒**Contextual bandit methods can be used for online algorithm selection**: ε-greedy, LinUCB[1], Randomized UCB[2], ILOVETOCONBANDITS[3]…

Methodology

Historical data (H): both the training data and the online data.

Instance	Feature 1	Feature 2	…	Algorithm 1	Algorithm 2	Algorithm 3
i_{train1}	10	T		10 sec	100 sec	50 sec
i_{train2}	20	F		1000 sec	50 sec	1000 sec
i_{train3}	15	F		100 sec	500 sec	50 sec
⋮						
$i_{online1}$	5	F		?	10 sec	?
$i_{online2}$	25	T		75 sec	?	?

Challenge: online data is incomplete; direct classification methods cannot use it
Solution: use regression-based methods (f.e. Algorithm 3)

Algorithm 1 Online algorithm selection

```
Input: training data H_T
Input: online strategy β
H = H_T
for instance i do
    λ = β(H) //Get selection map, based on all data
    a = λ(i) //Make selection
    Solve i with a, observing performance p
    H = H ∪ {i, φ, a, p} //Add newly generated data
```

Algorithm 3 Greedy online algorithm selection strategy

```
Input: H
∀a ∈ A : train regression model for a, based on the relevant records in H. p_{a,φ} is the resulting performance prediction for feature value vector φ
Define λ : λ(i) = arg max_{a∈A} p_{a,φ}
Return λ
```

Challenge: exploration vs. exploitation trade-off.
Solution: use strategies that explore (multi-armed bandit inspired)

- UCB-variant ($\lambda(i) = \arg\max_{a \in A}(p_{a,\varphi} + \lambda * sd_{a,\varphi})$)
- ε-greedy (random with probability ε, greedy otherwise)

Empirical study

Simulation study on 18 **ASlib** scenarios
Instance split: 10% train, 80% online, 10%verification
Results

- Offline learns decent models (> single best)
  ⇒Training on 10% of data already beneficial
- Greedy>Offline in 15 of 18 scenarios
- Greedy>exploring strategies always
  ⇒Cost of exploring insufficiently compensated
- Exploring strategies do not learn better models
  ⇒Exploration not beneficial here?

scenario	Improvement Online	Improvement Verif	Gap bridged Online	Gap bridged Verif
ASP-POTASSCO	-0.014	0.133	-6.24%	40.23%
BNSL-2016	0.063	0.094	33.91%	51.12%
CPMP-2015	0.033	0.051	34.11%	16.23%
CSP-2010	0.369	0.700	68.27%	85.86%
CSP-MZN-2013	0.037	0.074	45.42%	73.15%
GRAPHS-2015	-0.065	0.087	-33.92%	30.29%
MAXSAT12-PMS	0.030	0.161	12.91%	52.48%
MAXSAT15-PMS	0.032	0.239	14.43%	71.98%
PROTEUS-2014	-0.007	0.024	-0.53%	1.82%
QBF-2011	0.081	0.134	35.76%	62.70%
QBF-2014	0.081	0.175	28.17%	55.70%
SAT11-HAND	0.033	0.108	8.08%	28.76%
SAT11-INDU	0.070	-0.086	23.56%	-9.19%
SAT11-RAND	0.025	0.023	28.66%	22.66%
SAT12-ALL	0.046	0.101	15.18%	43.12%
SAT12-HAND	0.040	0.193	10.21%	37.92%
SAT12-INDU	0.023	0.127	7.40%	42.73%
SAT15-INDU	0.054	0.147	7.70%	11.69%

Model quality evolution over time, averaged over all scenarios

Ignores cost of exploration

Some observations

- Individual regr models are very inaccurate …but they rank algorithms quite well
- Normalising performance in function of VBS and single best can lead to extremely negative values when they are similar
  ⇒ Can bias result-aggregation

Related Work

- Malitsky's **eISAC**[4]: requires generation of additional data
- Gagliolo and Schmidhuber[5]: consider general **timeshare allocation**; model it as a standard bandit problem with as arms different time-allocators.
- Non-contextual bandits have been successfully applied to **intelligently switch algorithms while solving a single instance** by f.e. Cicerillo and Smith[6], and Lagoudakis and Littman[7], but without inter-instance knowledge transfer
- Misir and Sebag[9] modelled algorithm selection as a **collaborative filtering problem**. Can also handle incomplete data. Mention possibility of online setting, but do not test in this setting.

Current + Future Work

- Investigate **why explicit exploration is not beneficial**
  - Too much training data? Probably not, similar results with less
  - Parameter tuning? Probably not; verified for ε-greedy
  - Bad exploration strategies? Perhaps
  - Improving regression models do not imply overall improvements? Perhaps
  - Greedy explores in a way? Perhaps
- Use methods from contextual bandit literature
- Investigate start-from-zero setting

Contact

Hans Degroote
KU Leuven, Department of CS
Email: Hans.Degroote@kuleuven.be

References

[1] Chu, Wei, et al. "Contextual bandits with linear payoff functions." International Conference on Artificial Intelligence and Statistics. 2011.
[2] Dudik, Miroslav, et al. "Efficient optimal learning for contextual bandits." *arXiv preprint arXiv:1106.2369* (2011).
[3] Agarwal, Alekh, et al. "Taming the monster: A fast and simple algorithm for contextual bandits." International Conference on Machine Learning. 2014.
[4] Malitsky, Yuri. "Evolving instance-specific algorithm configuration." Instance-Specific Algorithm Configuration. Springer International Publishing, 2014. 93-105.
[5] 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.
[6] Cicirello, Vincent A., and Stephen F. Smith. "The max k-armed bandit: A new model of exploration applied to search heuristic selection." AAAI. 2005.
[7] Lagoudakis, Michail G., and Michael L. Littman. "Algorithm Selection using Reinforcement Learning." ICML. 2000.
[8] Bischl, Bernd, et al. "Aslib: A benchmark library for algorithm selection." Artificial Intelligence 237 (2016): 41-58
[9] Misir, Mustafa, and Michele Sebag. "Algorithm selection as a collaborative filtering problem." (2013): 43.