Solve Machine Learning with... Machine Learning!

Effectiveness of the Metropolis-Hastings algorithm for synthesizing Machine Learning Pipelines

1. Pipeline synthesis

Automatically generate a machine learning pipeline for given problem

Why?

Makes ML accessible to many more people!

2. Problem and our contribution

- the amount of possible pipelines is very large
- so an efficient search algorithm is needed
- we compare the performance of different search algorithms in the context of pipeline synthesis

3. Benchmark of ML problems

To compare the search algorithms, we create a dataset of 19 diverse ML problems

Name	ID	Entries	Features	Target classes
iris	61	150	4	3
seeds	1499	210	7	3
blood-transfusion	1464	748	4	2
diabetes	37	768	8	2
ilpd	1480	583	10	2
qsar-biodeg	1494	1.1k	41	2
monks-problems-2	334	601	6	2
tic-tac-toe	50	958	9	2
gas-drift	1476	13.9k	128	6
musk	1116	6.6k	167	2
madelon	1485	2.6k	500	2
gisette	41026	7.0k	5.0k	2
har	1478	10.3k	561	6
glass	41	214	9	6
car-evaluation	40664	1.7k	21	4
wdbc	1510	569	30	2
spambase	44	4.6k	57	2
wine-quality-red	40691	1.6k	11	6
wine-quality-white	40498	4.9k	11	7

4. Context-free grammar (CFG)

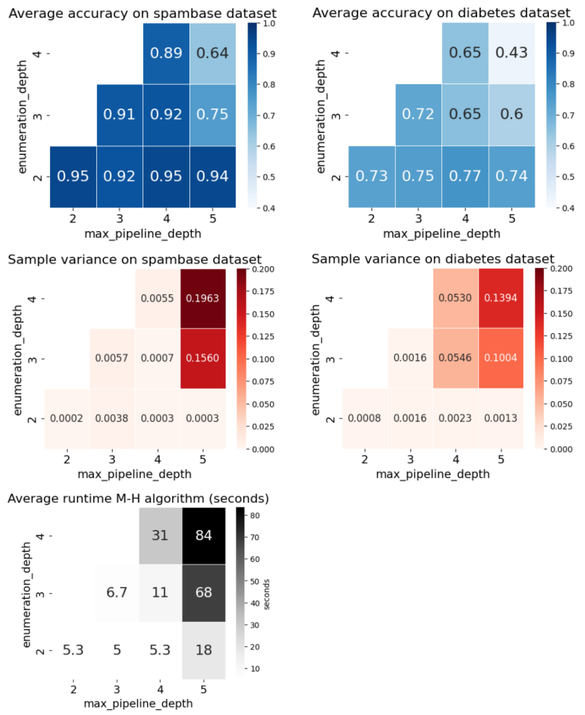
The search space of possible pipelines is defined by a CFG

```
<start>    ->  <classif> | seq(<pre>, <classif>)
<pre>      ->  <preproc> | <fselect> | seq(<pre>, <pre>) |
               par(<branch>, <branch>)
<branch>   ->  <pre> | <classif> | seq(<pre>, <classif>)

<preproc>  ->  StandardScaler | Binarizer | PCA | ...
<fselect>  ->  SelectKBest | SelectPercentile | ...
<classif>  ->  DecisionTreeClassifier |
               RandomForestClassifier | ...
```

5. Metropolis-Hastings algorithm

We test the algorithm with different hyperparameters to find the best combination



6. Evaluation and Results

We compare the average accuracy on three datasets from the benchmark

Algorithm	seeds	wdbc	har
BFS2	0.931	0.969	0.980
BFS4	0.925	0.949	0.982
M-H	0.919	0.959	0.969
VLSN	0.906	0.949	0.980
GA	0.847	0.912	0.760
MCTS	0.928	0.970	0.981
A*	0.919	0.965	0.970

Surprisingly, none of the more complicated search algorithms perform better than the simplest possible algorithm (BFS2)

7. Conclusions

- On simple datasets, naïve search algorithms work best
- Further experiments needed on more complex datasets

8. Get in touch

Project by: Denys Sheremet

Supervisors: Sebastijan Dumančić and Tilman Hinnerichs

Email author: denys.sheremet@gmail.com

Full paper available at: repository.tudelft.nl