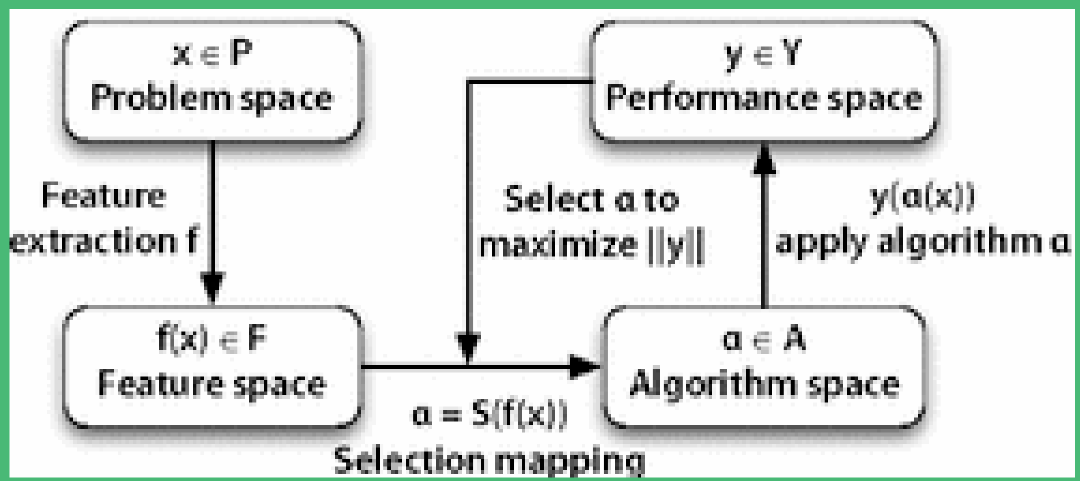
Algorithm Selection with continuous feature optimal decision trees

An adaption of ConTree's algorithm for instance cost-sensitive classification

Saunaq Chakrabarty, Koos v.d Linden, Emir Demirovic

TU Delft

How to decide which algorithm is best for a problem?



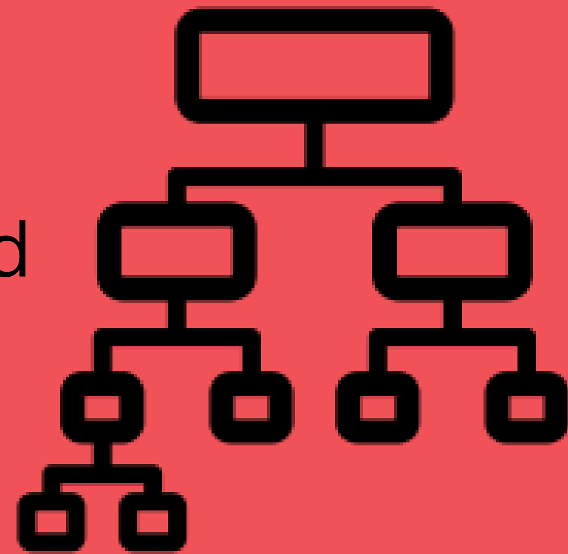
Many features influence algorithm performance

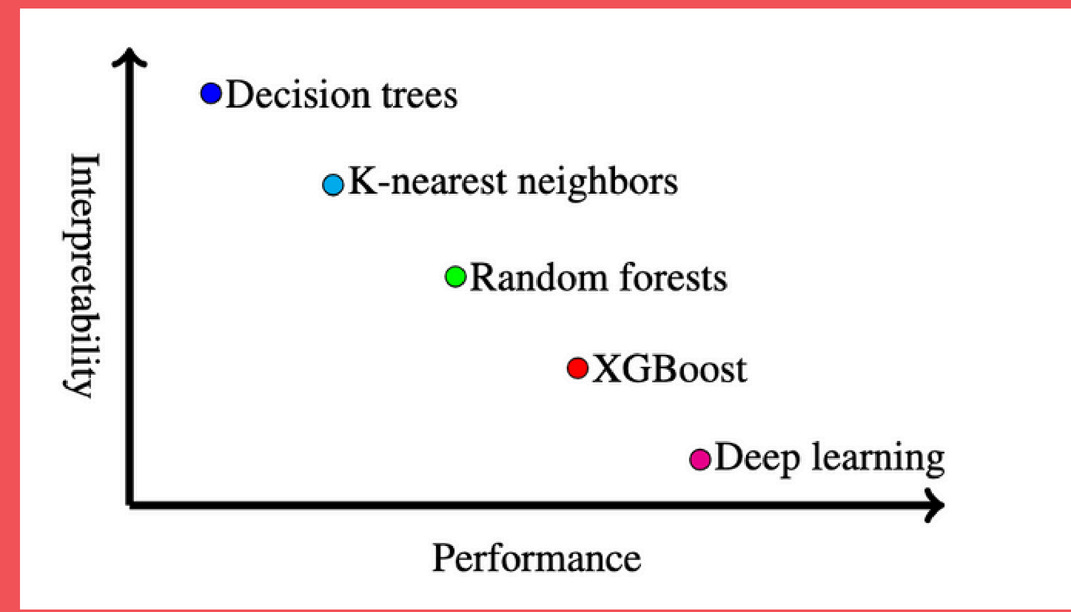
Algorithm Selection!

Goal: Find the best performing algorithm for a problem without running it.

$$\arg\min \sum_{x \in \mathcal{X}} \mathcal{M}(x, s(x)))$$

Decision Trees give an excellent balance between interpretability and accuracy





Unfortunately, traditional decision tree algorithms don't provably optimize the evaluation objective...

MurTree(binary features) and **ConTree**(continuous features) correct this by provably minimizing the misclassification score using DP

Research Question: Can we train an continuous feature optimal decision tree for the algorithm selection problem?

Method: Change the ConTree algorithm to optimize the normalized PAR10 score rather than the misclassification score

$$\mathcal{M}'(p,s) = \begin{cases} \mathcal{M}(p,s) & \text{if } \mathcal{M}(p,s) \le T \\ 10T & \text{else} \end{cases}$$

Contribution:

- Proved a new lower bound for the objective
- Adapted ConTree's pruning methods for this new lower bound
- Adapted the Depth Two Solver
- Added a regularization term to prevent overfitting.

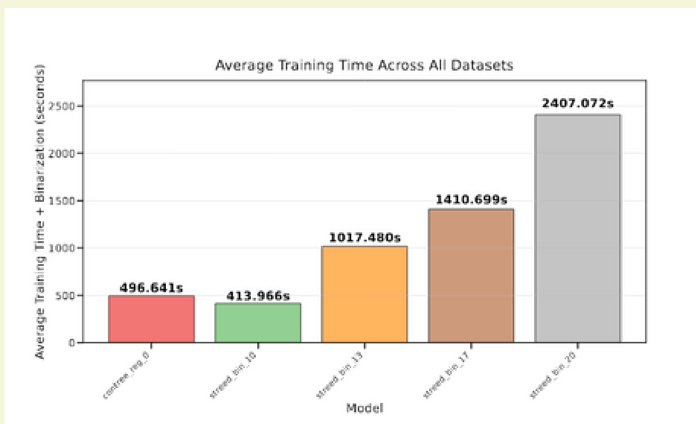
Results:

Comparable o.o.s PAR10 to state of the art methods

Scenario Name	ConTree++	Random Forest	STreeD
ASP-POTASSCO	0.76	**0.84**	0.76
CSP-2010	0.88	0.82	**0.94**
BNSL-2016	0.77	**0.86**	0.78
CPMP-2015	0.19	**0.36**	0.29
GLUHACK-2018	**0.28**	0.17	0.19
MAXSAT-2012PMS	**0.90**	0.89	0.87
PROTEUS-2014	0.68	**0.70**	**0.70**
QBF2016	**0.58**	0.41	0.5

Far better scalability than other methods for continuous features

Scenario Name	ConTree++	MIP
ASP-POTASSCO	**0.82**	-
CSP-2010	**0.06**	-
BNSL-2016	**0.33**	-
CPMP-2015	**0.007**	-
GLUHACK-2018	**0.03**	-
MAXSAT-2012PMS	**0.03**	-
PROTEUS-2014	**1.16**	-
QBF2016	**0.27**	-



Better scalability than binarized methods for sufficient binarization values

Limitations:

- Needs more robust experiments on scalability
- Less fast than binarized methods for low bin values

Future work

Tighter lower bounds and pruning to help scalability

More robust experiments on interpretability tradeoff

Contact:

saunaq002@gmail.com