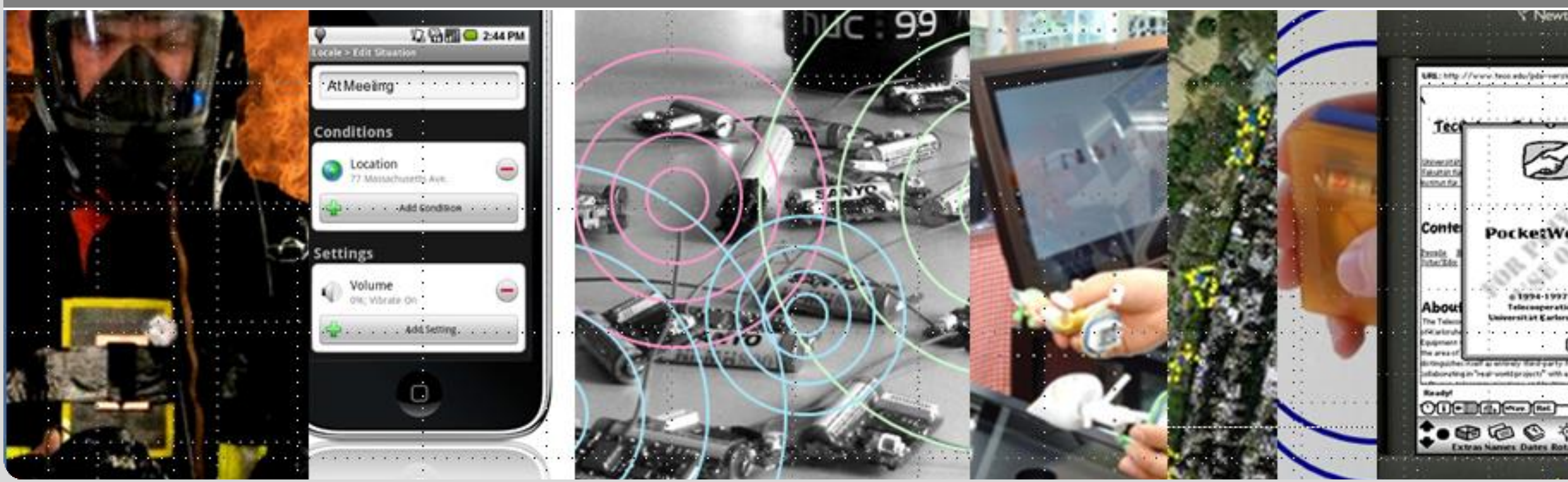
Karlsruhe Institute of Technology

Technology for Pervasive Computing

Title: AutoML with Monte Carlo Tree Search and Neural Network

Author: Shaohang Chen
Supervisor : Yiran Huang

CHAIR FOR PERVASIVE COMPUTER SYSTEMS, INSTITUTE OF TELEMATICS, DEPARTMENT OF COMPUTER SCIENCE



0. Design

- AutoML
- Monte Carlo Tree Search
- MCTML Structure
- MCTML Details
- Evaluation
- Future Work

1. AutoML

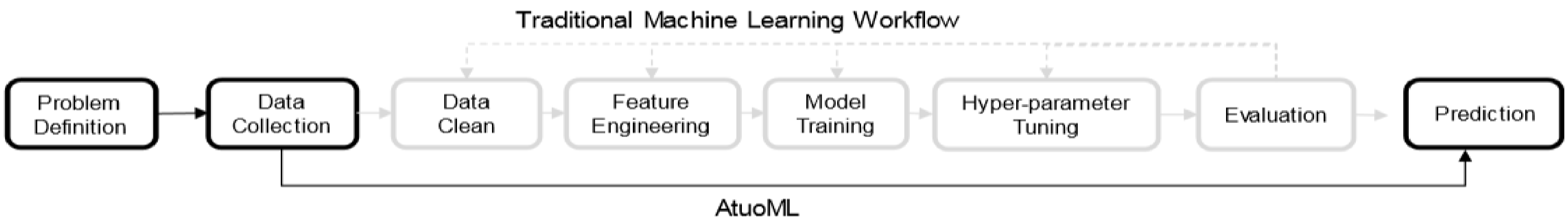


Figure 1.1: Traditional machine learning workflow and AutoML

Task: Algorithm selection and hyperparameter selection --- black box optimization[1]

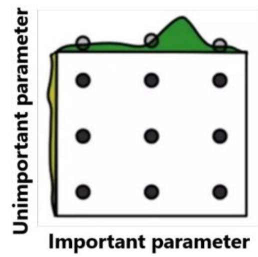


Figure 1.2: Grid Search

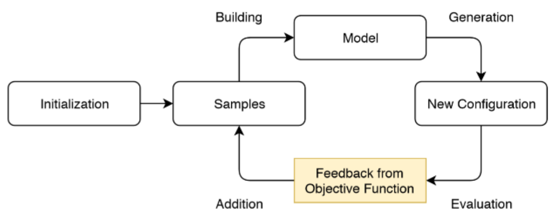


Figure 1.4: Baysian Optimization

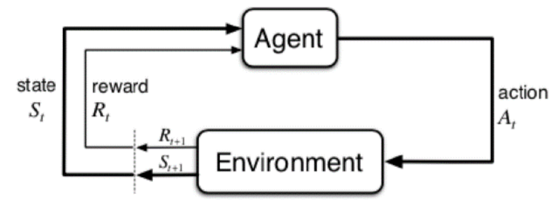


Figure 1.5: Reinforcement Learning

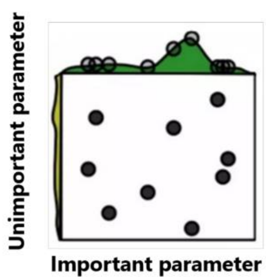


Figure 1.3: Random Search

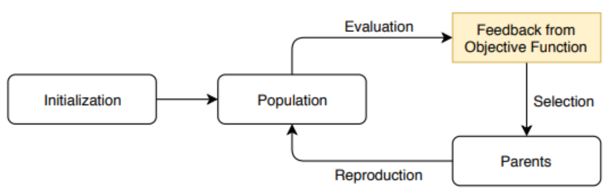


Figure 1.5: Genetic Algorithm

Method	Tool
Baysian Optimization	Autosklearn[2]
Genetic Algorithm	TPOT[3]
Reinforcement Learning	——

Table 1: AutoML methods and tools

[1] Automated Machine Learning: Methods, Systems, Challenges, Hutter, Frank, 2018
[2] Efficient and Robust Automated Machine Learning, Matthias Feurer, Aaron Klein, Katharina Eggensperger,2015
[3] TPOT: A Tree-Based Pipeline Optimization Tool for Automating Machine Learning, Randal S. Olson,etc, 2019

2. Monte Carlo Tree Search

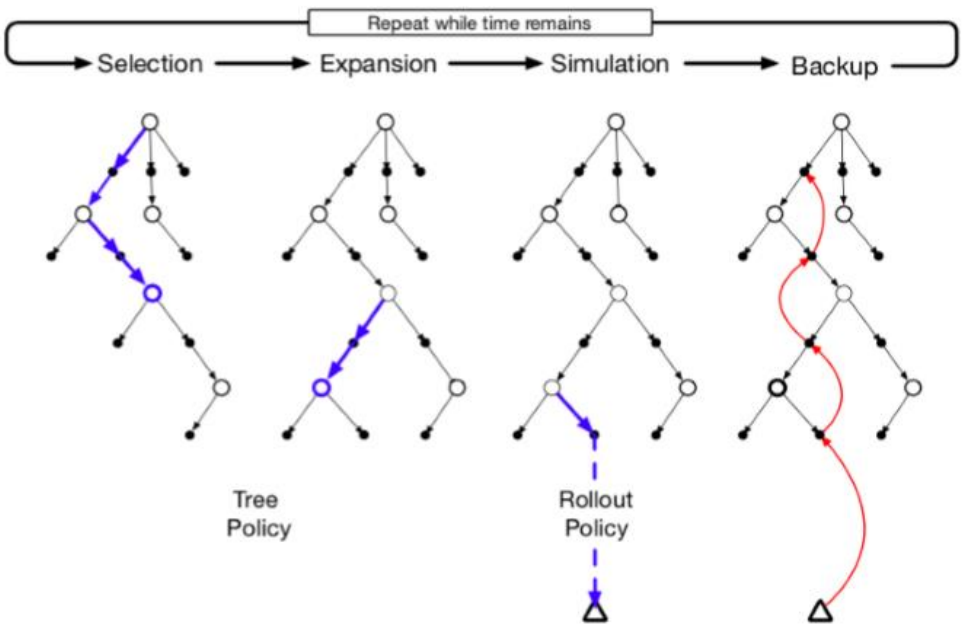


Figure 2.1: Outline of Monte Carlo Tree Search[1]

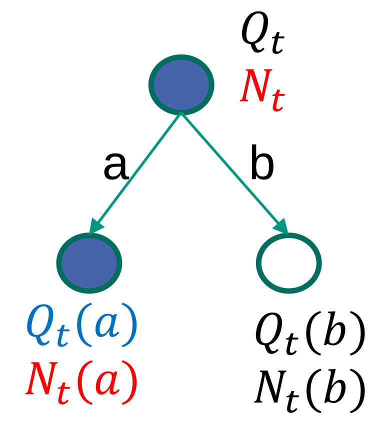


Figure 2.2: Ilussion of parent node and children nodes

$$A_t \doteq \underset{a}{\operatorname{argmax}}\left[Q_t(a) + c\sqrt{\frac{N_t}{N_t(a)}}\right]$$

Equation 1: Upper Confidence Bound Algorithm.

Selection: traverses and selects the best-scored child node(UCB algorithm)
Expansion: a new child node is added to the tree
Simulation: choosing moves until a leaf node is achieved
Backup: backprogress the value to each node in trajectory.

[1] Mastering the game of Go with deep neural networksand tree search, David Silver, Aja Huang, 2016

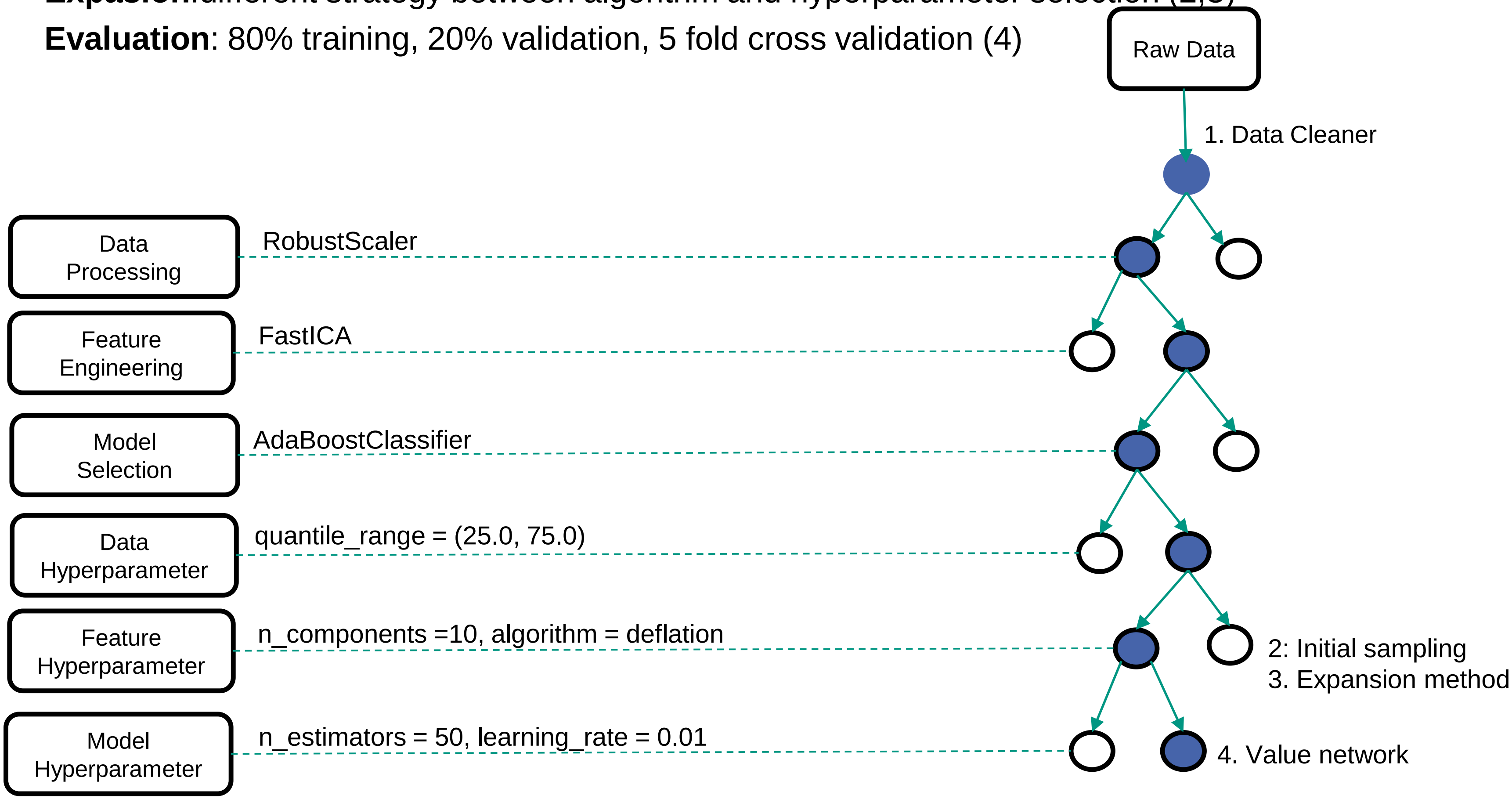
3.1. MCTML

- Structure

  Data Cleaning (1)

  Expasion:different strategy between algorithm and hyperparameter selection (2,3)

  Evaluation: 80% training, 20% validation, 5 fold cross validation (4)



3.2. Detail of MCTML(1)

- 1. Data cleaner
  - missing value imputation: NaN, ?, -
  - conver non-numerical feature to numerical feature

- 2. Initial sampling

hyperparameter name	value range	sampled values
n_estimators	(50,500)	[50,100,200,300,400,500]
learning_rate	(0.01,2.0)	[0.01,0.05,0.1,0.5,1.0,1.5,2.0]

Table 2 : Example of hyper-parameter samples using initial state sampling method

3.3. Detail of MCTML(2)

3. Expansion algorithm

New parameter: $K = \lceil C_{pw} \, N^{\alpha} \rceil$

When to expand node? All node visited at least 10, and K increase by one

Which node to expand? Selection between neighbors and random nodes.

4. Value network

Parameter Name	Value
Neuron Number of input layer	6
Neuron Number of hidden layer 1	10
Neuron Number of hidden layer 2	8
Neuron number of hidden layer 3	4
number of Neuron output layer	1
Optimizer	Adam
Number of epochs	100
Performance function	MSE
Training Goal	0.01

Table 3:Value network parameter setting

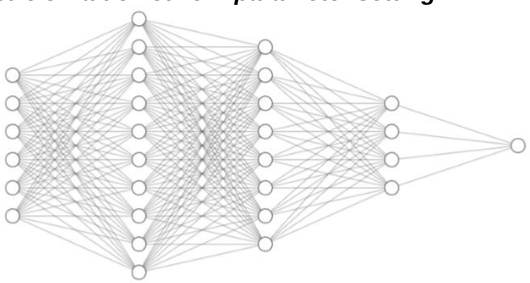


Figure 3.2: Value network structure

Algorithm 2 Expansion algorithm

Input: Node N
Output: new action A_{sug}

function PROGRESSIVE WIDENING(N)
    for $t = 0, 1, 2, 3...$ **do**
        Let $k = \lceil c_{pw} t^{\alpha} \rceil$, $A_{best} = \max_n Q_n$, $Q_{mean} = \frac{\sum_{n=1}^{n=N} Q_n}{N}$
        if $N_{Node} > 10$ and k increase by 1 **then**
            Find 2 neighbors A_{n1}, A_{n2}, 2 random samples A_{s1}, A_{s2}.
            choose the best evaluated sample A_{sug} and its evaluation $Q_{A_{sug}}$.
            if $Q_{A_{sug}} > Q_{mean}$ **then**
                return A_{sug}
            end if
        else
            return A_{best}
        end if
    end for
end function

Figure 3.1: Expansion Algorithm

4.1. Evaluation Environment

- Smart Data Innovation Lab Platform: 1 core with 4GB RAM.

- Search space: 5/5/14 (53)

  Data processor: 5

  Feature Engineering : 5

  Model: 14

  Hyperparameter: 53

- Dataset: OpenML CC18 suite

Name	Sample	Features	Class	Missing value
texture	5500	40	11	-
segment	2310	19	7	-
wilt	4839	6	2	-
car	1728	6	4	X
PhishingWebsites	11055	30	2	-

Table 3: Example of datasets properties from CC18 suite.

Name	Step	$\#\lambda$
quantile	Data	2
norm	Data	1
minmax	Data	1
robust	Data	1
stand	Data	-
Fastica	Feature	3
PCA	Feature	2
Polynomial	Feature	3
Agglomeration	Feature	1
KernelPCA	Feature	1
AdaBoost	Model	3
Bernoulli	Model	2
decision tree	Model	3
extra tree	Model	2
Gaussian naive Bayes	Model	1
Gradient boosting	Model	4
KNN	Model	3
LDA	Model	3
Linear SVC	Model	2
SVC	Model	3
Multinomial naive Bayes	Model	2
Passive aggressive	Model	4
QDA	Model	2
random forest	Model	4

Table 4 : Default search space, # means number of hyperparameters

4.2. Evaluation Results

Karlsruhe Institute of Technology

- Default setting

	Baseline	AutoSklearn	MCTML
Candidate models	SGD	Search Space	Search Space
Default parameter	learning rate = 0.0001 penalty = l2 loss function = hinge	ensemble size=1 meta learning = 0	c = 2 $c_{pw} = 4$ $\alpha = 0.3$
Variable parameter	—	total time = 30min	total time = 30min

Table 5 : Default setting of methods Baseline, AutoSklearn and MCTML

- Performance: $F1 = 2 * \frac{Precision\ * Recall}{Presion + Recall}$
- Stability: 5 runs variance
- Evaluation summary

	Baseline	AutoSklearn	MCTML
Performance	0	14	16
Stability	—	12	22

Table 6: Statistical summary of performance and stability

Dataset	MCTML	AutoSklearn
car	**0.986**	0.938
churn	**0.987**	0.335
climate	**0.975**	0.842
cnae-9	**0.792**	0.569
cylinder-bands	0.632	**0.831**
balance-scale	0.944	0.962
fourier	**0.921**	0.802
karhunen	**0.950**	0.658
satimage	**0.899**	0.766
morphological	**0.840**	0.752
eucalyptus	**0.905**	0.662
zernike	**0.818**	0.735
optdigits	**0.980**	0.948
pendigits	**0.982**	0.750
diabetes	**0.952**	0.463
spambase	**0.934**	0.851
splice	**0.095**	0.629
tic-tac-toe	**0.972**	0.488
letter	**0.889**	0.748
dna	**0.588**	0.331
first-order	**0.518**	0.490
jm1	0.466	**0.518**
kc1	0.326	**0.413**
kc2	0.419	0.435
plxel	**0.892**	0.811
MiceProtein	**0.991**	0.877
phpJNxH0q	**0.954**	0.897
ozone-level-8hr	**0.818**	0.666
pc1	0.286	0.306
pc3	0.195	**0.425**
PhishingWebsites	**0.925**	0.905
qsar-biodeg	**0.868**	0.631
segment	**0.957**	0.900
semeion	0.812	**0.836**
wall-robot	**0.923**	0.839
wdbc	0.963	0.941
wilt	**0.975**	0.857

Table 7: F1 score on all dataset of methods MCTML and Autosklearn

Technology for Pervasive Computing

5. Future Work

- Regression tasks --- change task type
- Initial state sampling --- meta learning
- Value network --- pipeline convertion and structure
- More evaluation methods & Bigger search space

Thanks