

Monte Carlo Tree Search for Algorithm Configuration: MOSAIC

Herilalaina Rakotoarison and Michèle Sebag

TAU

CNRS – INRIA – LRI – Université Paris-Sud



NeurIPS MetaLearning Wshop – Dec. 8, 2018

Monte Carlo Tree Search for Algorithm Configuration: MOSAIC

Herilalaina Rakotoarison and Michèle Sebag
Tackling the Underspecified

CNRS – INRIA – LRI – Université Paris-Sud



NeurIPS MetaLearning Wshop – Dec. 8, 2018

AutoML: Algorithm Selection and Configuration

A mixed optimization problem

$$\text{Find } \lambda^* \in \arg \min_{\lambda \in \Lambda} \mathcal{L}(\lambda, P)$$

with λ a pipeline and \mathcal{L} the predictive loss on dataset P

Modes

- ▶ offline hyper-parameter setting
- ▶ online hyper-parameter setting

Approaches

- ▶ Bayesian optimization: SMAC, Auto-SkLearn, AutoWeka, BHOB
Hutter et al., 11; Feurer et al. 15; Kotthoff et al. 17; Falkner et al. 18
- ▶ Evolutionary Computation Olson et al. 16; Choromanski et al. 18
- ▶ Bilevel optimization Franceschi et al. 17, 18
- ▶ Reinforcement learning Andrychowicz 16; Drori et al. 18

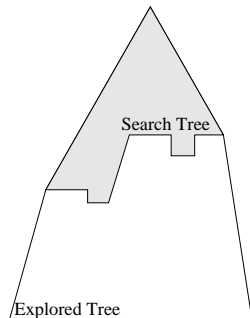
Monte Carlo Tree Search

Kocsis & Szepesvári 06, Gelly & Silver 07

Game playing when no good evaluation function and huge search space.

- ▶ Upper Confidence Tree (UCT)
 - ▶ Gradually grow the search tree
 - ▶ Building Blocks
 - ▶ Select next action (bandit-based phase)
 - ▶ Add a node (leaf of the search tree)
 - ▶ Select next action bis (random phase)
 - ▶ Compute instant reward
 - ▶ Update information in visited nodes
 - ▶ Returned solution
 - ▶ Path visited most often

Auer et al. 02



Within learning

Feature selection
Active learning

Gaudel, Sebag, 10
Rolet, Teytaud, Sebag, 09

Monte Carlo Tree Search

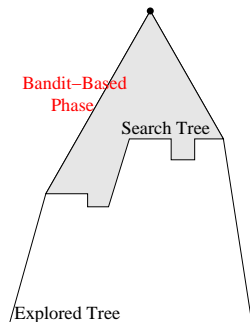
Kocsis & Szepesvári 06, Gelly & Silver 07

Game playing when no good evaluation function and huge search space.

▶ Upper Confidence Tree (UCT)

- ▶ Gradually grow the search tree
- ▶ Building Blocks
 - ▶ Select next action (bandit-based phase)
 - ▶ Add a node (leaf of the search tree)
 - ▶ Select next action bis (random phase)
 - ▶ Compute instant reward
 - ▶ Update information in visited nodes
- ▶ Returned solution
 - ▶ Path visited most often

Auer et al. 02



Within learning

Feature selection
Active learning

Gaudel, Sebag, 10
Rolet, Teytaud, Sebag, 09

Monte Carlo Tree Search

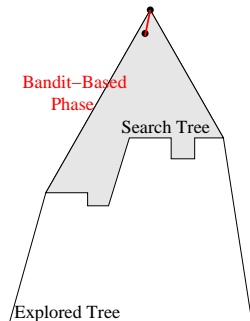
Kocsis & Szepesvári 06, Gelly & Silver 07

Game playing when no good evaluation function and huge search space.

▶ Upper Confidence Tree (UCT)

- ▶ Gradually grow the search tree
- ▶ Building Blocks
 - ▶ Select next action (bandit-based phase)
 - ▶ Add a node (leaf of the search tree)
 - ▶ Select next action bis (random phase)
 - ▶ Compute instant reward
 - ▶ Update information in visited nodes
- ▶ Returned solution
 - ▶ Path visited most often

Auer et al. 02



Within learning

Feature selection
Active learning

Gaudel, Sebag, 10
Rolet, Teytaud, Sebag, 09

Monte Carlo Tree Search

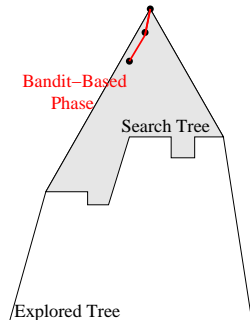
Kocsis & Szepesvári 06, Gelly & Silver 07

Game playing when no good evaluation function and huge search space.

▶ Upper Confidence Tree (UCT)

- ▶ Gradually grow the search tree
- ▶ Building Blocks
 - ▶ Select next action (bandit-based phase)
 - ▶ Add a node (leaf of the search tree)
 - ▶ Select next action bis (random phase)
 - ▶ Compute instant reward
 - ▶ Update information in visited nodes
- ▶ Returned solution
 - ▶ Path visited most often

Auer et al. 02



Within learning

Feature selection
Active learning

Gaudel, Sebag, 10
Rolet, Teytaud, Sebag, 09

Monte Carlo Tree Search

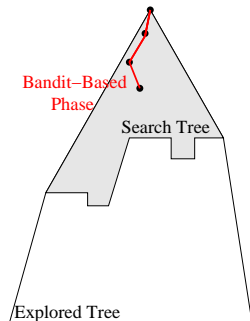
Kocsis & Szepesvári 06, Gelly & Silver 07

Game playing when no good evaluation function and huge search space.

▶ Upper Confidence Tree (UCT)

- ▶ Gradually grow the search tree
- ▶ Building Blocks
 - ▶ Select next action (bandit-based phase)
 - ▶ Add a node (leaf of the search tree)
 - ▶ Select next action bis (random phase)
 - ▶ Compute instant reward
 - ▶ Update information in visited nodes
- ▶ Returned solution
 - ▶ Path visited most often

Auer et al. 02



Within learning

Feature selection
Active learning

Gaudel, Sebag, 10
Rolet, Teytaud, Sebag, 09

Monte Carlo Tree Search

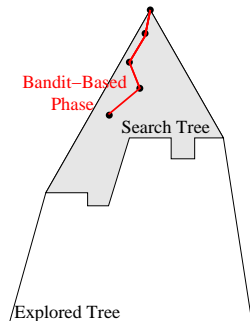
Kocsis & Szepesvári 06, Gelly & Silver 07

Game playing when no good evaluation function and huge search space.

▶ Upper Confidence Tree (UCT)

- ▶ Gradually grow the search tree
- ▶ Building Blocks
 - ▶ Select next action (bandit-based phase)
 - ▶ Add a node (leaf of the search tree)
 - ▶ Select next action bis (random phase)
 - ▶ Compute instant reward
 - ▶ Update information in visited nodes
- ▶ Returned solution
 - ▶ Path visited most often

Auer et al. 02



Within learning

Feature selection
Active learning

Gaudel, Sebag, 10
Rolet, Teytaud, Sebag, 09

Monte Carlo Tree Search

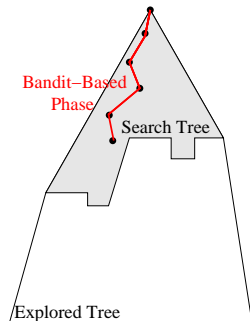
Kocsis & Szepesvári 06, Gelly & Silver 07

Game playing when no good evaluation function and huge search space.

▶ Upper Confidence Tree (UCT)

- ▶ Gradually grow the search tree
- ▶ Building Blocks
 - ▶ Select next action (bandit-based phase)
 - ▶ Add a node (leaf of the search tree)
 - ▶ Select next action bis (random phase)
 - ▶ Compute instant reward
 - ▶ Update information in visited nodes
- ▶ Returned solution
 - ▶ Path visited most often

Auer et al. 02



Within learning

Feature selection
Active learning

Gaudel, Sebag, 10
Rolet, Teytaud, Sebag, 09

Monte Carlo Tree Search

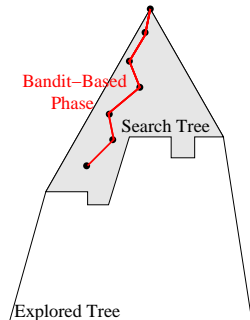
Kocsis & Szepesvári 06, Gelly & Silver 07

Game playing when no good evaluation function and huge search space.

▶ Upper Confidence Tree (UCT)

- ▶ Gradually grow the search tree
- ▶ Building Blocks
 - ▶ Select next action (bandit-based phase)
 - ▶ Add a node (leaf of the search tree)
 - ▶ Select next action bis (random phase)
 - ▶ Compute instant reward
 - ▶ Update information in visited nodes
- ▶ Returned solution
 - ▶ Path visited most often

Auer et al. 02



Within learning

Feature selection
Active learning

Gaudel, Sebag, 10
Rolet, Teytaud, Sebag, 09

Monte Carlo Tree Search

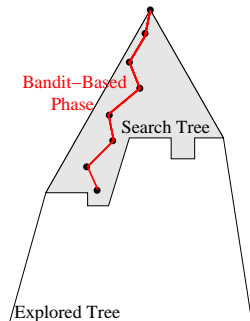
Kocsis & Szepesvári 06, Gelly & Silver 07

Game playing when no good evaluation function and huge search space.

▶ Upper Confidence Tree (UCT)

- ▶ Gradually grow the search tree
- ▶ Building Blocks
 - ▶ Select next action (bandit-based phase)
 - ▶ Add a node (leaf of the search tree)
 - ▶ Select next action bis (random phase)
 - ▶ Compute instant reward
 - ▶ Update information in visited nodes
- ▶ Returned solution
 - ▶ Path visited most often

Auer et al. 02



Within learning

Feature selection
Active learning

Gaudel, Sebag, 10
Rolet, Teytaud, Sebag, 09

Monte Carlo Tree Search

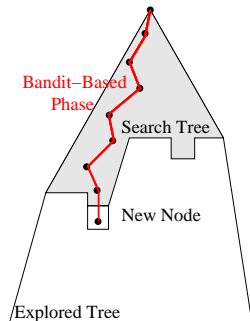
Kocsis & Szepesvári 06, Gelly & Silver 07

Game playing when no good evaluation function and huge search space.

▶ Upper Confidence Tree (UCT)

- ▶ Gradually grow the search tree
- ▶ Building Blocks
 - ▶ Select next action (bandit-based phase)
 - ▶ Add a node (leaf of the search tree)
 - ▶ Select next action bis (random phase)
 - ▶ Compute instant reward
 - ▶ Update information in visited nodes
- ▶ Returned solution
 - ▶ Path visited most often

Auer et al. 02



Within learning

Feature selection
Active learning

Gaudel, Sebag, 10
Rolet, Teytaud, Sebag, 09

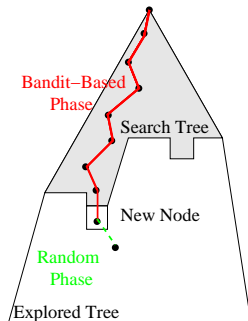
Monte Carlo Tree Search

Kocsis & Szepesvári 06, Gelly & Silver 07

Game playing when no good evaluation function and huge search space.

- ▶ Upper Confidence Tree (UCT)
 - ▶ Gradually grow the search tree
 - ▶ Building Blocks
 - ▶ Select next action (bandit-based phase)
 - ▶ Add a node (leaf of the search tree)
 - ▶ Select next action bis (random phase)
 - ▶ Compute instant reward
 - ▶ Update information in visited nodes
 - ▶ Returned solution
 - ▶ Path visited most often

Auer et al. 02



Within learning

Feature selection
Active learning

Gaudel, Sebag, 10
Rolet, Teytaud, Sebag, 09

Monte Carlo Tree Search

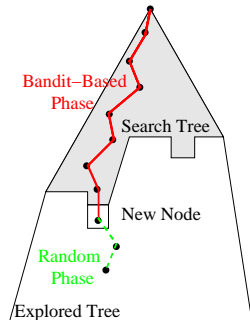
Kocsis & Szepesvári 06, Gelly & Silver 07

Game playing when no good evaluation function and huge search space.

▶ Upper Confidence Tree (UCT)

- ▶ Gradually grow the search tree
- ▶ Building Blocks
 - ▶ Select next action (bandit-based phase)
 - ▶ Add a node (leaf of the search tree)
 - ▶ Select next action bis (random phase)
 - ▶ Compute instant reward
 - ▶ Update information in visited nodes
- ▶ Returned solution
 - ▶ Path visited most often

Auer et al. 02



Within learning

Feature selection
Active learning

Gaudel, Sebag, 10
Rolet, Teytaud, Sebag, 09

Monte Carlo Tree Search

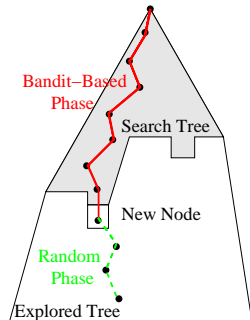
Kocsis & Szepesvári 06, Gelly & Silver 07

Game playing when no good evaluation function and huge search space.

▶ Upper Confidence Tree (UCT)

- ▶ Gradually grow the search tree
- ▶ Building Blocks
 - ▶ Select next action (bandit-based phase)
 - ▶ Add a node (leaf of the search tree)
 - ▶ Select next action bis (random phase)
 - ▶ Compute instant reward
 - ▶ Update information in visited nodes
- ▶ Returned solution
 - ▶ Path visited most often

Auer et al. 02



Within learning

Feature selection
Active learning

Gaudel, Sebag, 10
Rolet, Teytaud, Sebag, 09

Monte Carlo Tree Search

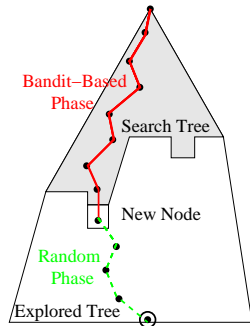
Kocsis & Szepesvári 06, Gelly & Silver 07

Game playing when no good evaluation function and huge search space.

▶ Upper Confidence Tree (UCT)

- ▶ Gradually grow the search tree
- ▶ Building Blocks
 - ▶ Select next action (bandit-based phase)
 - ▶ Add a node (leaf of the search tree)
 - ▶ Select next action bis (random phase)
 - ▶ Compute instant reward
 - ▶ Update information in visited nodes
- ▶ Returned solution
 - ▶ Path visited most often

Auer et al. 02



Within learning

Feature selection
Active learning

Gaudel, Sebag, 10
Rolet, Teytaud, Sebag, 09

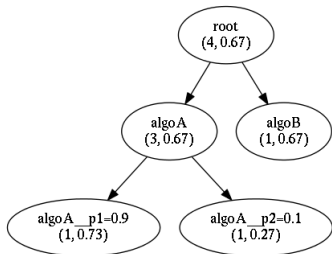
Monte Carlo Tree Search for AutoML

1. **Select** next action (alg/hyperparameter)

$$\text{select } \arg \max_i \left\{ \mu_i + c \sqrt{\frac{\log N}{n_i}} \right\}$$

with μ_i average reward, n_i number visits,
 $N = \sum_i n_i$

2. **Add a node**: new alg or hyper-parameter;
3. **Random phase**: complete pipeline with default/random choices.
4. **Compute reward** v : predictive accuracy of pipeline
5. Use v to **update** μ_i , increment n_i in all visited nodes



Mosaic: MCTS for AutoML

Overview

- ▶ Search space: $\{ \text{Preprocessing algs} \} \times \{ \text{Algorithms} \}$
 - ▶ Fixed sequence of choices:
 1. Preprocessing alg
 2. hyper-parameters of pre-processing alg
 3. Algorithm
 4. hyper-parameters of Alg.
- PCA, random proj, ...
dimension
SVM, RF, ...
C, ϵ , ...

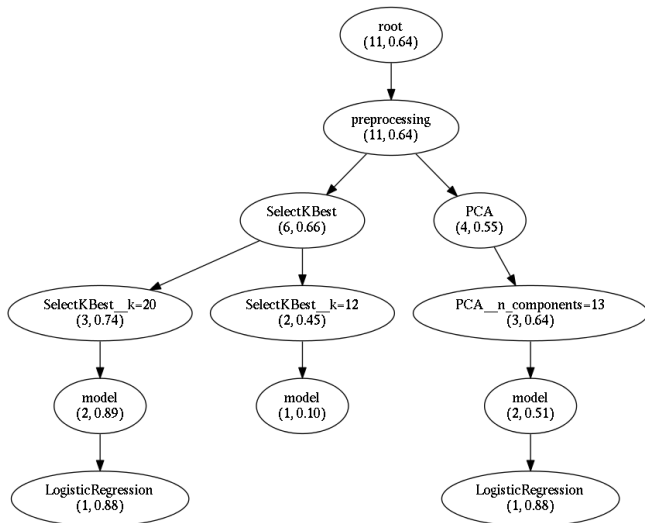
Key features

- ▶ CPU management
(every Δt , kills unpromising pipelines; increases evaluation resources for others)
- ▶ Sample continuous hyperparameters: Progressive widening

Gelly Silver 07

Increase number of sampled values like $\lfloor \sqrt{N} \rfloor$

A MOSAIC Tree



Search space

	Methods	Parameters
Preprocessing	PCA	n_components
	SelectKBest	k, score_func
	Gaussian Random Projection	n_components, eps
	No preprocessing	-
Algorithm	Logistic regression	C, penalty, solver
	SGD Classifier	learning rate, penalty, alpha, l1 ratio, loss
	KNN classifier	K, metric, weights
	XGBoost classifier	learning rate, max depth, gamma, subsample, regularization
	LDA	n_components, learning decay
	Random forest	criterion, max features, max depth, bootstrap, min sample split

Experiments and Results

AutoML Challenge PAKDD 2018

- ▶ Binary classification (Final phase)
- ▶ 10 or 20 minutes time budget for each dataset
- ▶ Metric: balanced accuracy

	Set 1	Set 2	Set 3	Set 4	Set 5
aad_freiburg	0.5533	0.2839	0.3932	0.2635	0.6766
mosaic	0.5382	0.3161	0.3376	0.3182	0.6317
narnars0	0.5418	0.2894	0.3665	0.2005	0.6922
W wang	0.5655	0.4851	0.2829	-0.0886	0.6840

Final test phase: mosaic ranked second w.r.t. average rank.

Experiment 2: extended pre-processing search space

Methods	Parameters
PCA	n_components, whiten, svd_solver, tol, iterated_power
KernelPCA	n_components, kernel, gamma, degree, coef0, alpha, eigen_solver, tol, max_iter
FastICA	n_components, algorithm, max_iter, tol, whiten, fun
Identity	-
IncrementalPCA	n_components, whiten, batch_size
SelectKBest	score_func, k
SelectPercentile	score_func, percentile
LinearSVC Pre-processing	C, class_weight, max_iter
ExtraTreesClassifier Pre-processing	n_estimators, criterion, max_depth, min_samples_split, min_samples_leaf, min_weight_fraction_leaf, max_features, max_leaf_nodes, class_weight
FeatureAgglomeration	n_clusters, affinity, linkage
PolynomialFeatures	degree
RBFSampler	gamma, n_components
RandomTreesEmbedding	n_components, max_depth, min_samples_split, min_samples_leaf, min_weight_fraction_leaf, max_leaf_nodes, min_impurity_decrease

Experiment 2: extended algorithm search space

Algorithms	Parameters
LinearDiscriminantAnalysis	solver, shrinkage
QuadraticDiscriminantAnalysis	reg_param
DummyClassifier	-
AdaBoostClassifier	base_estimator, n_estimators, learning_rate, algorithm
ExtraTreesClassifier	n_estimators, criterion, max_depth, min_samples_split, min_samples_leaf, min_weight_fraction_leaf, max_features, max_leaf_nodes, class_weight
RandomForestClassifier	n_estimators, criterion, min_samples_split, min_samples_leaf, min_weight_fraction_leaf, max_features, max_leaf_nodes, class_weight, bootstrap
GradientBoostingClassifier	loss, learning_rate, n_estimators, max_depth, criterion, min_samples_split, min_samples_leaf, min_weight_fraction_leaf, subsample, max_features, max_leaf_nodes
SGD Classifier	learning rate, penalty, alpha, l1 ratio, loss, epsilon, eta0, power_t, class_weight, max_iter

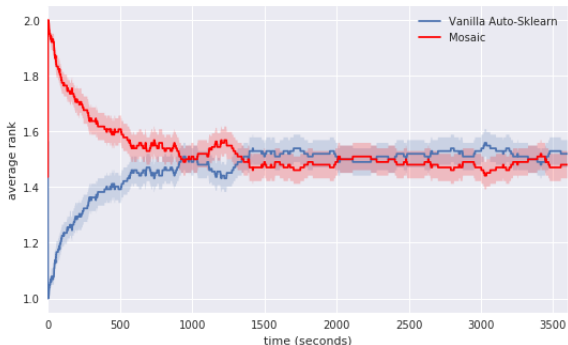
Experiment 2: extended algorithm search space, foll.

Algorithms	Parameters
Perceptron	penalty, alpha, max_iter, tol, shuffle, eta0
RidgeClassifier	alpha, max_iter, class_weight, solver
PassiveAggressiveClassifier	C, max_iter, tol, loss, class_weight
KNeighborsClassifier	n_neighbors, weights, algorithm, leaf_size, p, metric
MLPClassifier	hidden_layer_sizes, activation, solver, alpha, batch_size, learning_rate, learning_rate_init, power_t, max_iter, shuffle, warm_start, momentum, nesterovs_momentum, early_stopping, validation_fraction, beta_1, beta_2, epsilon
SVC	C, max_iter, tol, loss, class_weight, kernel, degree, gamma, coef0
DecisionTreeClassifier	criterion, splitter, max_depth, min_samples_split, min_samples_leaf, min_weight_fraction_leaf, max_features, max_leaf_nodes, min_impurity_decrease, class_weight
ExtraTreeClassifier	criterion, splitter, max_depth, min_samples_split, min_samples_leaf, min_weight_fraction_leaf, max_features, max_leaf_nodes, min_impurity_decrease, class_weight

Experiment 2

On 133 datasets from the OpenML repository
1 hour per (dataset, run); 10 runs per dataset.

Vanschoren et al. 14



Average rank (lower is better) of MOSAIC and Vanilla Auto-Sklearn across 102 datasets (Datasets on which the performance of both methods differs statistically according to Mann-Whitney rank test with $p = 0.05$).

Discussion

Monte Carlo Tree Search for Algorithm Configuration

- ▶ Proof of concept

Limitations

- ▶ Order of hyper-parameters
- ▶ Time allocation

Perspectives

Short term

- ▶ Refine initialization
- ▶ Extend to constrained satisfaction

Medium term

- ▶ Learn value of parameters across datasets
- ▶ Improving the sampling of continuous hyperparameter values

Long term

- ▶ Learning Meta-Features: a (the) key Meta-Learning task.