ITMO UNIVERSITY

Reinforcement-based Simultaneous Classification Model and its Hyperparameters Selection

Valeria Efimova, vefimova@corp.ifmo.ru

Andrey Filchenkov, afilchenkov@corp.ifmo.ru

Anatoly Shalyto, shalyto@mail.ifmo.ru

IDP-16.

Barcelona, Spain, 10–14 Oct 2016

Motivation

- Many algorithms for classification problem
- Most of an algorithms have hyperparameters
- We should select algorithm and its hyperparameters for a given dataset

Notation

Supervised learning problem:

- Input is a learning set D_{train}= $\{d_1, d_2, \ldots, d_n\}$, where $d_i = (x_i, y_i) \in X \times Y$.
- The goal is to approximate dependency with $f\colon X \to Y$, where X is feature set, Y is label set.

A learning algorithm A finds an f for each D.

- Λ^A is hyperparameter space related to A.
- A_λ is the algorithm A with chosen hyperparameter vector $\lambda \in \Lambda^A$.

Subproblem of algorithm selection

Input: a dataset D, a set of algorithms with hyperparameters $\mathscr{A} = \left\{A^1_{\lambda_1}, A^2_{\lambda_2}, \dots A^n_{\lambda_n}\right\}$.

The goal is to find the most efficient A_λ.

Quality measure is hold-out empirical risk $Q\left(A_\lambda, D\right)$.

Algorithm selection problem: $A^*_{\lambda_*} \in \underset{A^i_{\lambda_i} \in \mathscr{A}}{\operatorname{argmin}} Q\left(A^i_{\lambda_i}, D\right)$

Subproblem of hyperparameter optimization

Input: an algorithm A and a dataset D.

The goal is to find the best hyperparameter vector from Λ^A.

Hyperparameter optimization problem: $\lambda^* \in \min\limits_{\lambda \in \Lambda^A} Q(A_\lambda, D)$.

Problem of a learning algorithm and its hyperparameters selection

Input: a dataset D, a set of algorithms $\mathscr{A} = \{A^1, A^2, \dots, A^n\}$, and hyperparameter spaces $\{\Lambda^i\}$ associated with $\{A^i\}$.

The goal is to find the best learning algorithm and its hyperparameters.

Algorithm and its hyperparameters simultaneous selection problem: $A^*_{\lambda_*} \in \underset{A^i \in \mathscr{A}, \lambda_i \in \Lambda^i}{\operatorname{argmin}} Q(A^i_{\lambda_i}, D)$.

Hyperparameter optimization process

Sequential hyperparameter optimization process π_i is associated with each learning algorithm A^i:

$$\pi_i\left(t, A^i, \{\lambda_j\}_{j=0}^{k}\right) \rightarrow \lambda_{k+1}^i \in \Lambda^i,$$

where t is a time limitation for hyperparameter optimization.

Algorithm and its hyperparameter selection within a certain time

We are given a time budget T.

Split it: T = $t_1 + t_2 + \ldots + t_q$, $t_i \geq 0$.

New minimization problem:

$$\min_i Q(A^i_{\lambda_i}, D) \xrightarrow{(t_1, t_2, \ldots, t_n)} \min,$$

where $\lambda_i = \pi_i \left(\mathrm{t}, A^i, \{\lambda_j\}_{j=0}^{i-1}\right)$.

Exploration and exploitation

- Only exploration (assigning time for tuning hyperparameters of different algorithms) results into less time for the best algorithm tuning
- Only exploitation (assigning time for tuning hyperparamters of the current best algorithm) results into missing better algorithms.
- A tradeoff should be found.

Reduction to a multi-armed bandit problem

- At each iteration k, an agent chooses an arm a_i and gets a reward $r(i, k)$.
- The agent's goal is to minimize total loss by time T.
- Arm a_i is a hyperparameter optimization process for algorithm A^i.
- Playing an arm a_i is running a process π_i.
- Naïve approach to define a reward function is the difference between current empirical risk and optimal empirical risk found during previous iterations.

SMAC details

- At each iteration, a set of current optimal hyperparameter vectors is known for each algorithm.
- A local search is applied to find hyperparameter vectors such that they have distinction in one position with an optimal vector and improve algorithm quality. These hyperparameter vectors are added to the set.
- Some random hyperparameter vectors are added in the set.
- Then selected configurations (the algorithms with their hyperparameters) are sorted by **Expected Improvement** (EI).
- Some of the best configurations are run after that.

The suggested reward function

- As in SMAC, we use empirical risk expectation at iteration k: $E_t(Q(A^i_{\lambda^i_k}, D))$, where $Q\left(A^i_{\lambda^i_k}, D\right)$ is empirical risk value reached by process π_i on dataset D at iteration k.
- Since process π_i optimizes hyperparameters for empirical risk minimization, but a multi-armed bandit problem is maximization problem. we define average reward function as:

$$\bar{r}_{i,(k)} = \frac{Q_{\max} - E_{(t)}\left(Q\left(A^i_{\lambda^i_k}, D\right)\right)}{Q_{\max}},$$

- where $Q_{\max}$ is the maximal empirical risk that can be achieved on given dataset.

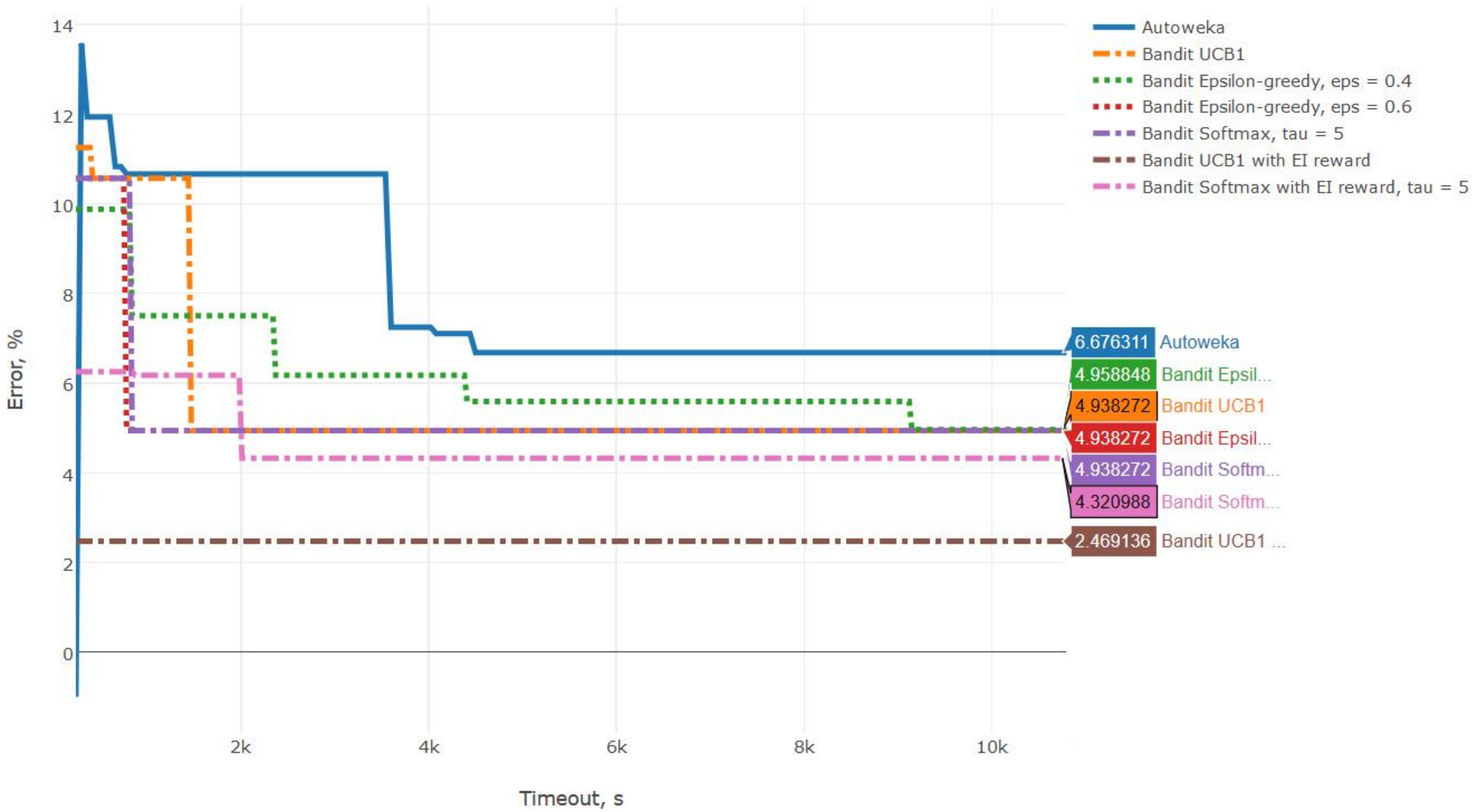
Multi-armed bandit algorithms

- ε-greedy
- UCB1
- Softmax

The experiments

- 6 classification algorithms: kNN, Support Vector Machine, Random Forest, Logistic Regression, Perceptron, C4.5 Decision Tree.
- 10 real datasets from UCI repository
- 6 versions of the suggested method: UCB1, 0.4-greedy, 0.6-greedy, Softmax with naïve reward function and UCB1, Softmax with the suggested reward function
- 12 runs
- 3 hours (10800 seconds) for each run
- 30 second per iteration
- SMAC hyperparameter optimization algorithm
- The comparison with Auto-WEKA library

The resulting graph for Dorothea dataset



The comparison with Auto-WEKA library

Наб. данных	AutoWEKA	UCB1	0,4-жадный	0,6-жадный	Softmax	$UCB1_{E(Q)}$	$Softmax_{E(Q)}$
Car	**0,3305**	0,373	0,3834	0,39	0,3734	0,3623	0,39
Yeast	36,79	35,41	35,33	35,52	35,36	**35,19**	35,36
KR-vs-KP	0,4138	0,3539	0,3576	0,3508	0,3520	**0,3495**	0,3539
Semeion	5,25	**3,475**	3,752	3,826	3,551	3,558	3,64
Shuttle	0,01719	0,01609	0,01625	0,1618	0,15	0,1452	**0,01438**
Dexter	10,71	5,04	4,943	5,193	**4,796**	**4,801**	4,877
Waveform	11,83	10,207	10,72	10,61	10,184	**9,886**	10,196
Secom	4,805	4,5	4,636	4,55	4,47	**4,432**	4,444
Dorothea	9,632	5,26	6,637	5,966	6,033	**4,869**	5,404
German Credits	20,6	18,3	18,27	19,69	18,24	**17,71**	17,92

Conclusion

- We suggested and evaluated a new solution for the actual problem of an algorithm and its hyperparameters simultaneous selection.
- The proposed approach is based on the multi-armed bandit problem solution.
- We suggested a reward function exploiting hyperparameter optimization method properties.
- The suggested method outperforms the existing method implemented in Auto-WEKA library.

ITMO UNIVERSITY

Thank you!

Reinforcement-based Simultaneous Classification Model and its Hyperparameters Selection

Valeria Efimova, Andrey Filchenkov, Anatoly Shalyto,

vefimova@corp.ifmo.ru, afilchenkov@corp.ifmo.ru, shalyto@mail.ifmo.ru