Supplementary material for AutoMMLC: An Automated and Multi-objective Method for Multi-label Classification

Aline Marques Del Valle[1,2], Rafael Gomes Mantovani[3], and Ricardo Cerri[2]

[1] Federal Inst. of Education, Science and Technology of the South of Minas Gerais
Campus of Muzambinho - MG - Brazil
aline.valle@ifsuldeminas.edu.br
[2] Department of Computer Science - Federal University of São Carlos
Rodovia Washington Luis, km 235 - São Carlos - SP - Brazil
cerri@ufscar.br
[3] Federal University of Technology – Parana (UTFPR),
Campus of Apucarana - PR - Brazil
rafaelmantovani@utfpr.edu.br,

1 Search Space

Automated Machine Learning (AutoML) aims to find good Machine Learning (ML) models automatically. Given the optimization algorithm, the search space, and the evaluation criterion(s), AutoML searches for the best model. The search space defines the set of possible algorithms and hyperparameters used during the search.

This document describes the search space used in the AutoML method proposed in this work, denominated the Automated Multi-objective Multi-label Classification (AutoMMLC). It was composed of Single-label Classification (SLC) and Multi-label Classification (MLC) algorithms that would use different hyperparameters. For each algorithm, we considered values of hyperparameters already mentioned in the literature. The hyperparameters were described in the terminology of the *scikit-multilearn* [12] and *scikit-learn* [7] libraries.

Table 1 presents the MLC algorithms search space. It is composed of the algorithms: Binary Relevance (BR), BR and KNN in version A (BRkNNa), BR and KNN in version B (BRkNNb), Classifier Chains (CC), Decision Tree (DT), Multi-Label Adaptive Resonance Associative Map (MLARAM), Multi-Label KNN (MLkNN), Multi-Label Support Vector Machines (MLTSVM), Random Forest (RF), and Random k labelsets Disjoint (RakelD).

Ranges of values represent the values of algorithm hyperparameters. In our work, intervals were represented by fixed-size vectors. For real numbers, for example, if the hyperparameter value is of 0.1 to 0.5, we predefine the range values ($[0.1, 0.2, 0.3, 0.4, 0.5]$) instead of considering all reals numbers between the lower and upper bounds. The hyperparameters *classifier* of the BR and CC algorithms and *base_classifier* of the RakelD algorithm have as possible values all the SLC algorithms listed in Table 2.

Table 2 brings the search space for the SLC algorithms. It is composed of the algorithms: AdaBoost, Bernoulli Naïve Bayes (BNB), Decision Tree (DT), Extra Tree (ET), K-Nearest Neighbors (KNN), Logistic Regression (LR), Multi-Layer Perceptron (MLP), Multinomial Naïve Bayes (MNB), Random Forest (RF), Stochastic Gradient Descent (SGD), and Support Vector Machine (SVM) algorithm.

For the SLC search space, we also changed the values of the hyperparameters about the values cited in the literature. For the AdaBoost algorithm, we considered 3 of the 5 hyperparameters mentioned by [8]. In the KNN algorithm, we considered only the odd values ranging from 1 to 100 for *n_neighbors*. We define the number of hidden layers for the MLP algorithm according to [9], where L detonates the number of labels and F the number of features. Finally, in the *loss* hyperparameter of the SGD algorithm, we only consider the *log* value.

Table 1. MLC Search Space containing multi-label algorithms and their hyperparameters. Hyperparameters are represented by arrays, which can contain numerical or nominal values.

Algorithm	**Hyperparameters**	**References**
BR	classifier: All algorithms SLC	-
BRkNNa	k: {1, 2, ... 30}	[11]
BRkNNb	k: {1, 2, ..., 30}	[11]
CC	classifier: All algorithms SLC	-
DT	criterion: {gini, entropy} max_depth: {1, 2, ..., 10} min_samples_split: {2, 3, ..., 20} min_samples_leaf: {1, 2, ..., 20}	[4]
MLARAM	vigilance: {0.001, 0.05, 0.1, ..., 0.9, 0.95} threshold: {0.001, 0.05, 0.1, ..., 0.9, 0.95}	[1,10]
MLkNN	k: {6, 8, 10 . . . 20}	[6]
MLTSVM	c_k: {2e-6, 2e-5, ..., 2e6} sor_omega: 0.2 lambda_param: {2e-4, 2e-3, ..., 2e4}	[1,2]
RakelD	base_classifier: All algorithms SLC labelset_size: 3	[13]
RF	criterion: {gini, entropy} max_features: {0.1, 0.2, ..., 1} min_samples_split: {2, 3, ..., 20} min_samples_leaf: {1, 2, ..., 20} bootstrap: {True, False}	[3]

Table 2. SLC Search Space. It contains single-label algorithms and hyperparameters represented by numerical or nominal arrays.

Algorithm	Hyperparameters	References
AdaBoost	learning_rate: {1e-3, 1e-2, 1e-1, 0.2, ..., 2.0} algorithm: {SAMME.R, SAMME} max_depth: {1, 2, ..., 10}	[8]
BNB	alpha: {1e-3, 1e-2, 1e-1, 1, 10, 100} fit_prior: {True, False}	[4]
DT	criterion: {gini, entropy} max_depth: {1, 2, ..., 10} min_samples_split: {2, 3, ..., 20} min_samples_leaf: {1, 2, ..., 20}	[4]
ET	criterion: {gini, entropy} max_features: {0.05, 0.1, ..., 1} min_samples_split: {2, 3, ..., 20} min_samples_leaf: {1, 2, ..., 20} bootstrap: {True, False}	[3]
KNN	n_neighbors: {1, 3, ..., 99} weights: {uniform, distance} p: {1, 2}	[4]
LR	penalty: l2 C: {1e-4, ..., 1e-1, 0.5, 1, 5, 10, ..., 25} dual: False solver: {newton-cg, lbfgs, sag, saga} max_iter: 3000	[5]
MLP	activation: {tanh, relu, logistic} alpha: {1e-7, 1e-6, ..., 1e-1} early_stopping: {True, False} learning_rate_init: {1e-4, ..., 1e-1, ..., 0.5} hidden_layer_sizes: {L, (L + F)/2, F, L + F} max_iter: 3000	[3,9]
MNB	alpha: {1e-3, 1e-2, 1e-1, 1, 10, 100} fit_prior: {True, False}	[4]
RF	criterion: {gini, entropy} max_features: {0.1, 0.2, ..., 1} min_samples_split: {2, 3, ..., 20} min_samples_leaf: {1, 2, ..., 20} bootstrap: {True, False}	[3]
SGD	alpha: {1e-7, 1e-6, ..., 1e-1} average: {True, False} epsilon: {1e-5, 1e-4, ..., 1e-1} eta0: {1e-7, 1e-6, ..., 1e-1} l1_ratio: {1e-9, 1e-8, ..., 1e-1, 0.5, 1} learning_rate: {optimal, invscaling, constant} loss: log penalty: {l1, l2, elasticnet} power_t: {1e-5, 1e-4, ..., 1e-1, 1} tol: {1e-5, 1e-4, ..., 1e-1}	[3]
SVM	kernel: rbf gamma: {2e-15, 2e-13, ..., 2e3} C : {2e-5, 2e-3, ..., 2e15}	[1]

References

1. Bogatinovski, J., Todorovski, L., Džeroski, S., Kocev, D.: Comprehensive comparative study of multi-label classification methods. Expert Systems with Applications **203**, 117215 (2022). https://doi.org/https://doi.org/10.1016/j.eswa.2022.117215
2. Chen, W.J., Shao, Y.H., Li, C.N., Deng, N.Y.: Mltsvm: A novel twin support vector machine to multi-label learning. Pattern Recognition **52**, 61–74 (2016). https://doi.org/https://doi.org/10.1016/j.patcog.2015.10.008
3. Feurer, M., Eggensperger, K., Falkner, S., Lindauer, M., Hutter, F.: Auto-sklearn 2.0: Hands-free automl via meta-learning. Journal of Machine Learning Research **23**(261), 1–61 (2022), `http://jmlr.org/papers/v23/21-0992.html`
4. Fu, W., Olson, R., Nathan, Jena, G., PGijsbers, Augspurger, T., Romano, J., Saha, P., Shah, S., Raschka, S., sohnam, DanKoretsky, kadarakos, Jaimeccclin, bartdp1, Bradway, G., Ortiz, J., Smit, J.J., Menke, J.H., Ficek, M., Varik, A., Chaves, A., Myatt, J., Ted, Badaracco, A.G., Kastner, C., Jerônimo, C., Hristo, Rocklin, M., Carnevale, R.: Epistasislab/tpot: v0.11.5 (Jun 2020). https://doi.org/10.5281/zenodo.3872281
5. Gijsbers, P., LeDell, E., Poirier, S., Thomas, J., Bischl, B., Vanschoren, J.: An open source automl benchmark (Jun 2019), `https://sites.google.com/view/automl2019icml/home`, 6th ICML Workshop on Automated Machine Learning, AutoML@ICML2019 ; Conference date: 14-06-2019 Through 14-06-2019
6. Madjarov, G., Kocev, D., Gjorgjevikj, D., Džeroski, S.: An extensive experimental comparison of methods for multi-label learning. Pattern Recognition **45**(9), 3084–3104 (2012). https://doi.org/https://doi.org/10.1016/j.patcog.2012.03.004, best Papers of Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA'2011)
7. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research **12**, 2825–2830 (2011)
8. van Rijn, J.N., Hutter, F.: Hyperparameter importance across datasets. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery amp; Data Mining. p. 2367–2376. KDD '18, Association for Computing Machinery, New York, NY, USA (2018). https://doi.org/10.1145/3219819.3220058
9. de Sá, A.G.C., Pimenta, C.G., Pappa, G.L., Freitas, A.A.: A robust experimental evaluation of automated multi-label classification methods. In: Proceedings of the 2020 Genetic and Evolutionary Computation Conference. p. 175–183. GECCO '20, Association for Computing Machinery, New York, NY, USA (2020). https://doi.org/10.1145/3377930.3390231
10. Sapozhnikova, E.P.: Art-based neural networks for multi-label classification. In: Adams, N.M., Robardet, C., Siebes, A., Boulicaut, J.F. (eds.) Advances in Intelligent Data Analysis VIII. pp. 167–177. Springer Berlin Heidelberg, Berlin, Heidelberg (2009)
11. Spyromitros, E., Tsoumakas, G., Vlahavas, I.: An empirical study of lazy multilabel classification algorithms. In: Darzentas, J., Vouros, G.A., Vosinakis, S., Arnellos, A. (eds.) Artificial Intelligence: Theories, Models and Applications. pp. 401–406. Springer Berlin Heidelberg, Berlin, Heidelberg (2008)
12. Szymanski, P., Kajdanowicz, T.: Scikit-multilearn: A scikit-based python environment for performing multi-label classification. J. Mach. Learn. Res. **20**(1), 209–230 (jan 2019)

13. Tsoumakas, G., Katakis, I., Vlahavas, I.: Random k-labelsets for multilabel classification. IEEE Transactions on Knowledge and Data Engineering **23**(7), 1079–1089 (2011). https://doi.org/10.1109/TKDE.2010.164