Annotated Bibliography

CASH: Combined Algorithm Selection and Hyperparameter optimization
The challenge of machine learning: given a dataset, to automatically and simultaneously choose a learning algorithm and set its hyperparameters to optimize empirical performance.

Bayesian Optimization: fits a probabilistic model to capture the relationship between hyperparameter settings and their measured performance; this model is then used to select the most promising hyperparameter setting, evaluates that hyperparameter setting, updates the model with the result, and iterates

AutoML Packages: Auto-WEKA (Java), hyperopt-sklearn, Auto-sklearn, TPOT, Auto-Keras, Auto-PyTorch

Thornton, C., Hutter, F., Hoos, H. H., & Leyton-Brown, K. (2013). Auto-WEKA: Combined Selection and Hyperparameter Optimization of Classification Algorithms. *ArXiv:1208.3719 [Cs]*. http://arxiv.org/abs/1208.3719

This paper explores an approach to simultaneous selection of ML algorithms and its hyperparameters using Bayesian optimization (Sequential Model-Based Optimization). The paper considers a wide range of feature selection techniques, a combination of 3 search and 8 evaluator methods and uses all the classification approaches used in WEKA. Which includes 2 ensemble methods, 10 meta-methods, 27 base classifiers and hyperparameter settings for each classifier. The classification performance of Auto-Weka was often better than using standard selection and hyperparameter optimization methods. In conclusion the paper did find that Auto-Weka had a bit of an overfitting problem as it showed larger improvements in cross-validation performance than on the test data and requires a much better method than the simple correlation-based approach used.

Abstract: Many different machine learning algorithms exist; taking into account each algorithm's hyperparameters, there is a staggeringly large number of possible alternatives overall. We consider the problem of simultaneously selecting a learning algorithm and setting its hyperparameters, going beyond previous work that addresses these issues in isolation. We show that this problem can be addressed by a fully automated approach, leveraging recent innovations in Bayesian optimization. Specifically, we consider a wide range of feature selection techniques (combining 3 search and 8 evaluator methods) and all classification approaches implemented in WEKA, spanning 2 ensemble methods, 10 meta-methods, 27 base classifiers, and hyperparameter settings for each classifier. On each of 21 popular datasets from the UCI repository, the KDD Cup 09, variants of the MNIST dataset and CIFAR-10, we show classification performance often much better than using standard selection/hyperparameter optimization methods. We hope that our approach will help non-expert users to more effectively identify machine learning algorithms and hyperparameter settings appropriate to their applications, and hence to achieve improved performance.

Kotthoff, L., Thornton, C., Hoos, H. H., Hutter, F., & Leyton-Brown, K. (2019). Auto-WEKA: Automatic Model Selection and Hyperparameter Optimization in WEKA. In F. Hutter, L. Kotthoff, & J. Vanschoren (Eds.), *Automated Machine Learning* (pp. 81–95). Springer International Publishing. https://doi.org/10.1007/978-3-030-05318-5_4

This paper introduces a new package that is integrated with WEKA, they found that although Bayesian optimization based on Gaussian process models is known to perform well for low-dimensional problems with numerical hyperparameters, tree-based models have been shown to be more effective for high-dimensional, structured, and partly discrete problems. Auto-WEKA 2.0 expands on Auto-WEKA by supporting regression algorithms, adding the optimization of all performance metrics WEKA supports and fully integrated with WEKA.

Abstract: WEKA is a widely used, open-source machine learning platform. Due to its intuitive interface, it is particularly popular with novice users. However, such users often find it hard to identify the best approach for their particular dataset among the many available. We describe the new version of Auto-WEKA, a system designed to help such users by automatically searching through the joint space of WEKA’s learning algorithms and their respective hyperparameter settings to maximize performance, using a state-of-the-art Bayesian optimization method. Our new package is tightly integrated with WEKA, making it just as accessible to end users as any other learning algorithm.

Feurer, M., Klein, A., Eggensperger, K., Springenberg, J., Blum, M., & Hutter, F. (n.d.). *Efficient and Robust Automated Machine Learning*. 9.

Auto-sklearn is a AutoML system based on scikit-learn which uses 15 classifiers, 14 feature preprocessing methods, and 4 data preprocessing methods, giving rise to a structured hypothesis space with 110 hyperparameters. This system improves on existing AutoML methods by considering past performance on similar datasets by using a case-based reasoning system called k-nearest datasets (KND). Attempts to solve the CASH problem like Auto-WEKA. To increase efficiency and robustness of AutoML, Auto-Sklearn includes a meta-learning step to warmstart the Bayesian optimization and an automated ensemble construction step. Auto-Sklearn also uses a the random-forest-based SMAC to solve the CASH problem in this paper.

Abstract: The success of machine learning in a broad range of applications has led to an ever-growing demand for machine learning systems that can be used off the shelf by non-experts. To be effective in practice, such systems need to automatically choose a good algorithm and feature preprocessing steps for a new dataset at hand, and also set their respective hyperparameters. Recent work has started to tackle this automated machine learning (AutoML) problem with the help of efficient Bayesian optimization methods. Building on this, we introduce a robust new AutoML system based on scikit-learn (using 15 classifiers, 14 feature preprocessing methods, and 4 data preprocessing methods, giving rise to a structured hypothesis space with 110 hyperparameters). This system, which we dub AUTO-SKLEARN, improves on existing AutoML methods by automatically taking into account past performance on similar datasets, and by constructing ensembles from the models evaluated during the optimization. Our system won the first phase of the ongoing ChaLearn AutoML challenge, and our comprehensive analysis on over 100 diverse datasets shows that it substantially outperforms the previous state of the art in AutoML. We also demonstrate the performance gains due to each of our contributions and derive insights into the effectiveness of the individual components of AUTO-SKLEARN.

Olson, R. S., & Moore, J. H. (2019). TPOT: A Tree-Based Pipeline Optimization Tool for Automating Machine Learning. In F. Hutter, L. Kotthoff, & J. Vanschoren (Eds.), *Automated Machine Learning* (pp. 151–160). Springer International Publishing. https://doi.org/10.1007/978-3-030-05318-5_8

TPOT is a genetic-based (using DEAP) AutoML system that optimizes the feature preprocessors with the goal of maximizing the accuracy of the classification tasks. TPOT is a wrapper for the scikit-learn package in python and automates feature selection, preprocessing as well as supervised classification of data.

Abstract: As data science becomes more mainstream, there will be an ever-growing demand for data science tools that are more accessible, flexible, and scalable. In response to this demand, automated machine learning (AutoML) researchers have begun building systems that auto- mate the process of designing and optimizing machine learning pipelines. In this paper we present TPOT v0.3, an open source genetic programming-based AutoML system that optimizes a series of feature preprocessors and machine learning models with the goal of maximizing classification accuracy on a supervised classification task. We benchmark TPOT on a series of 150 supervised classification tasks and find that it significantly outperforms a basic machine learning analysis in 21 of them, while experiencing minimal degradation in accuracy on 4 of the benchmarks—all without any domain knowledge nor human input. As such, GP-based AutoML systems show considerable promise in the AutoML domain.

Feurer, M., Eggensperger, K., Falkner, S., Lindauer, M., & Hutter, F. (2021). Auto-Sklearn 2.0: Hands-free AutoML via Meta-Learning. *ArXiv:2007.04074 [Cs, Stat]*. http://arxiv.org/abs/2007.04074

This paper details the improvements made to Auto-sklearn and discusses the benefits of each change. Improvements were made to the model selection strategy, portfolio building and automated policy selection. These improvements provide an entirely hands-free system, which seamlessly provides the best setup given a new task and resource limitations.

Abstract: Automated Machine Learning, which supports practitioners and researchers with the tedious task of manually designing machine learning pipelines, has recently achieved substantial success. In this paper we introduce new Automated Machine Learning (AutoML) techniques motivated by our winning submission to the second ChaLearn AutoML challenge, PoSH Auto-sklearn. For this, we extend Auto-sklearn with a new, simpler meta-learning technique, improve its way of handling iterative algorithms and enhance it with a successful bandit strategy for budget allocation. Furthermore, we go one step further and study the design space of AutoML itself and propose a solution towards truly hand-free AutoML. Together, these changes give rise to the next generation of our AutoML system, Auto-sklearn (2.0). We verify the improvement by these additions in a large experimental study on 39 AutoML benchmark datasets and conclude the paper by comparing to Auto-sklearn (1.0), reducing the regret by up to a factor of five.

Barros, R. C., Basgalupp, M. P., Freitas, A. A., & de Carvalho, A. C. P. L. F. (2014). Evolutionary Design of Decision-Tree Algorithms Tailored to Microarray Gene Expression Data Sets. *IEEE Transactions on Evolutionary Computation*, *18*(6), 873–892. https://doi.org/10.1109/TEVC.2013.2291813

Abstract: Decision-tree induction algorithms are widely used in machine learning applications in which the goal is to extract knowledge from data and present it in a graphically intuitive way. The most successful strategy for inducing decision trees is the greedy top-down recursive approach, which has been continuously improved by researchers over the past 40 years. In this paper, we propose a paradigm shift in the research of decision trees: instead of proposing a new manually designed method for inducing decision trees, we propose automatically designing decision-tree induction algorithms tailored to a specific type of classification data set (or application domain). Follow- ing recent breakthroughs in the automatic design of machine learning algorithms, we propose a hyper-heuristic evolutionary algorithm called hyper-heuristic evolutionary algorithm for de- signing decision-tree algorithms (HEAD-DT) that evolves design components of top-down decision-tree induction algorithms. By the end of the evolution, we expect HEAD-DT to generate a new and possibly better decision-tree algorithm for a given application domain. We perform extensive experiments in 35 real-world microarray gene expression data sets to assess the performance of HEAD-DT, and compare it with very well known decision- tree algorithms such as C4.5, CART, and REPTree. Results show that HEAD-DT is capable of generating algorithms that significantly outperform the baseline manually designed decision- tree algorithms regarding predictive accuracy and F-measure.

de Sá, A. G. C., Pinto, W. J. G. S., Oliveira, L. O. V. B., & Pappa, G. L. (2017). RECIPE: A Grammar-Based Framework for Automatically Evolving Classification Pipelines. In J. McDermott, M. Castelli, L. Sekanina, E. Haasdijk, & P. García-Sánchez (Eds.), *Genetic Programming* (Vol. 10196, pp. 246–261). Springer International Publishing. https://doi.org/10.1007/978-3-319-55696-3_16

Abstract: Automatic Machine Learning is a growing area of machine learning that has a similar objective to the area of hyper-heuristics: to automatically recommend optimized pipelines, algorithms or appropriate parameters to specific tasks without much dependency on user knowledge. The background knowledge required to solve the task at hand is actually embedded into a search mechanism that builds personalized solutions to the task. Following this idea, this paper proposes RECIPE (REsilient ClassifIcation Pipeline Evolution), a framework based on grammar-based genetic programming that builds customized classification pipelines. The framework is flexible enough to receive different grammars and can be easily extended to other machine learning tasks. RECIPE overcomes the drawbacks of previous evolutionary-based frameworks, such as generating invalid individuals, and organizes a high number of possible suitable data pre-processing and classification methods into a grammar. Results of f-measure obtained by RECIPE are compared to those two state-of-the-art methods, and shown to be as good as or better than those previously reported in the literature. RECIPE represents a first step towards a complete framework for dealing with different machine learning tasks with the minimum required human intervention.