A Hybrid Approach for Automatic Model Recommendation

Roman Vainshtein
Ben-Gurion University of the Negev
Beer-Sheva, Israel
romanva@post.bgu.ac.il

Asnat Greenstein-Messica
Ben-Gurion University of the Negev
Beer-Sheva, Israel
asnatm@post.bgu.ac.il

Gilad Katz
Ben-Gurion University of the Negev
Beer-Sheva, Israel
giladkz@post.bgu.ac.il

Bracha Shapira
Ben-Gurion University of the Negev
Beer-Sheva, Israel
bshapira@post.bgu.ac.il

Lior Rokach
Ben-Gurion University of the Negev
Beer-Sheva, Israel
liorrk@post.bgu.ac.il

ABSTRACT

One of the challenges of automating machine learning applications is the automatic selection of an algorithmic model for a given problem. We present AutoDi, a novel and resource-efficient approach for model selection. Our approach combines two sources of information: metafeatures extracted from the data itself and word-embedding features extracted from a large corpus of academic publications. This hybrid approach enables AutoDi to select top-performing algorithms both for widely and rarely used datasets by utilizing its two types of feature sets. We demonstrate the effectiveness of our proposed approach on a large dataset of 119 datasets and 179 classification algorithms grouped into 17 families. We show that AutoDi can reach an average of 98.8% of optimal accuracy and select the optimal classification algorithm in 49.5% of all cases.

CCS CONCEPTS

• **Computing methodologies** → **Machine learning approaches**; *Information extraction*; • **Theory of computation** → Machine learning theory;

KEYWORDS

Classification, Classifier Families ,Meta-Learning, Dataset Metafeatures ,Scholarly Big Data, Algorithm Recommendation, Word Embedding, Expert System

ACM Reference Format:
Roman Vainshtein, Asnat Greenstein-Messica, Gilad Katz, Bracha Shapira, and Lior Rokach. 2018. A Hybrid Approach for Automatic Model Recommendation. In *The 27th ACM International Conference on Information and Knowledge Management (CIKM '18), October 22–26, 2018, Torino, Italy*. ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3269206.3269299

1 INTRODUCTION

The explosion of digital data has created a wealth of opportunities for individuals and organizations alike, enabling them to gain insights that were not previously possible. As machine learning is often needed to take advantages of these opportunities, its use has become widespread. Since many of the new users of machine learning are non-experts, they require assistance in tasks such as selecting the appropriate machine learning algorithms and setting their parameters. While there are existing solutions (notably AutoWeka [13] and Auto-sklearn [4]) applying them requires a large amount of time and computing resources.

We present AutoDi, an efficient method for algorithm recommendation. Based on the intuition that similar types of problems can be addressed by the same algorithms, we extract metafeatures that represent various characteristics of the dataset and use word embeddings to model the type of challenges it poses. We use these metafeatures to train a ranking model that returns a list of algorithms, sorted by their predicted efficacy on the data.

In addition to its high accuracy (98.8% of the optimal performance in our experiments), a key advantage of AutoDi is that it is resource-efficient and fast. By training our recommendation meta-model "offline" we are able to output recommendations for new datasets without training any models on the analyzed dataset. In addition to producing high-quality results on its own, our approach can be considered as a pre-filtering step for more computationally-expensive algorithm and hyperparameter selection methods, such as Auto-Weka [13].

2 RELATED WORK

2.1 Meta-Learning for Algorithm Selection

One of the first works that defined Meta-Learning as the study of principled methods that exploit meta-knowledge (meta-features) to obtain efficient models and solutions by adapting machine learning and data mining processes was done by Brazdil et al. [2]. The literature groups meta-features into three types: 1) statistical 2) model-based and 3) landmarkers [1]. In the statistical group we can find the number of examples of the dataset, attribute correlation, class entropy etc. Such application provides both metafeatures and interpretable knowledge about the problem [2].

Model-based meta-features Peng et al. [8] capture characteristics of a model generated by applying learning algorithms to dataset, e.g., the number of leaf nodes of a decision tree. Finally, landmarkers [4] are generated by making a quick performance estimate of a simple learning algorithm in the dataset. Pfahringer et al. [10]. One of the latest works by Fabio Pinto et al. [11] presents a framework to systematically generate features for Meta-Learning. The

CIKM '18, October 22–26, 2018, Torino, Italy

ACM ISBN 978-1-4503-6014-2/18/10…$15.00
https://doi.org/10.1145/3269206.3269299

researchers showed that the sets of generated metafeatures are more informative than both the non-systematic ones and the state-of-the-art. Recent studies [4, 13] focuses on iterative and simultaneous selection of learning algorithms and hyperparameters tuning.

2.2 Embedding Representation of Knowledge

Word embedding methods, such as Word2Vec [7] and GloVe [9] attracted a great amount of attention in the recent years. The vector representation of words learned by these methods has been shown to carry semantic meanings and is useful in various natural language process (NLP) tasks. These methods produced state-of-the-art performance in word analogy such as "king is to queen as man is to woman" and similarity tasks. Furthermore, these vectors, when used as the underlying input representation, have been shown to boost the performance in various NLP applications In this research we trained a GloVe model on academic publication to model both algorithms and types of classification challenges.

3 APPROACH

We propose a hybrid model for the recommendation of classification algorithms for previously unseen datasets. Our approach utilizes two sources of information: a) meta-features representing the analyzed dataset and; b) word embeddings extracted from hundreds of thousands of academic publications. These embeddings model both the problem represented by each dataset and the algorithms used to address it. Our rationale for combining the two approaches is as follows: for new datasets whose characteristics are similar to many of the ones we previously analyzed, it is reasonable to hypothesize that the meta-features used to describe the datasets will provide most of the needed insights to effectively recommend an algorithm. For novel datasets or datasets with uncommon characteristics, the vast body of academic work represented by the embedding model is likely to be the more effective approach.

3.1 Dataset-based metafeatures

Our dataset-based meta-features are designed to capture various aspects of the analyzed dataset that are likely to be relevant to the statistical nature of machine learning algorithms. Our features are based on the works of Pinto et al. [11] and Katz et al. [5], who used this type of feature to enhance the performance of ML algorithms. Our list comprises of 21 metafeatures which can be divided into the following groups:

- **General statistics** – number of instances, missing values, number of features, number of numerical and discrete features etc.
- **Summary statistics** – statistics on the value distributions of the various features of the datasets. These features include averages, standard deviations and entropy calculations of the various dataset features.
- **Correlation-based statistics** – the goal of these features is to estimate how correlated are the various features of the dataset. The features of this group include the means and standard deviations of the Spearman and Pearson correlation calculations.

3.2 Embedding-based features

The goal of our embedding-based features is to take advantage of the extensive body of work produced by human researchers over the years. We hypothesize that many of these studies describe state-of-the-art solutions (at least at the time the paper was published) and can therefore serve as guidelines for algorithm selection.

Generating the term embedding lists

We generate our word embeddings using the following process:

- To obtain a large number of papers, we crawl the Engineering Village website, a large repository of academic papers which offers access to 13 databases of engineering literature and patents. Overall, we downloaded 461,420 academic papers.
- We applied the GloVe framework [9] to generate word embeddings from the extracted papers. Word embeddings do not only generate a more compact representation terms compared to one-hot-dot encoding, but also enable us to model contextual relationships among terms . In addition, under the assumption that newer academic work will produce superior results, we modified the GloVe algorithm to give greater weight to recent academic work.
- Using the titles of the pages under the "algorithms" and "applications" categories in Wikipedia, we compiled lists of terms representing classification algorithms and classification problems respectively. These terms of each list were matched to the relevant vectors in the word embedding.

Upon its completion, the process described above produces two lists of terms – one of algorithms and one of problems – where each term is represented by an embedding vector. We use these terms for algorithms recommendation.

Recommending algorithms for a given dataset

Since our embeddings enable us to recommend a set of algorithms for a given problem description, we require that every dataset we analyze have a short description of the challenge it represents and/or its features. This requirement proved easy to satisfy since almost every dataset in open repositories such as UCI and OpenML possesses such a description (see section 4.1 for details). We then perform the following steps:

- We identify all the terms in the "problems" list that appear in the description.
- For each term in the "algorithms" list, we calculate its Cosine similarity to each of the detected "problems" terms and sum the values. This is done using the following formula:

$$S_m = \sum_{i \in D} \cos(w_m, w_i)$$

  where w_m represents the model keyword embedded vector, w_i represents the embedded vector for each dataset matched term, and D is the set of all matched dataset keywords.
- Finally, we produce a sorted list of the the terms in the "algorithms" list based on their final score, where algorithms that received higher scores are deemed more likely to perform well on the analyzed dataset.

3.3 Generating the hybrid model

We evaluated two approaches for combining the dataset and embedding-based features. In the *weighted averaging* approach, we trained two separate models, each using a different group of metafeatures, and then used weighted averaging to combine the scores of the two models. In the second approach, which we denote the *sequential*

approach, we provide the score of embedding-based model as an additional feature to the model trained on the dataset-based metafeatures. We now elaborate on the two approaches:

The weighted-averaging approach. In this approach we trained two separate models, one for each of the two groups of features. For the dataset-based metafeatures, we used to XGBoost algorithm to predict the actual performance (i.e. the accuracy) of each of the possible classification algorithms. The values of this algorithm were therefore in the range [0,1]. For the embedding-based features, the score of each algorithm was calculated using the formula presented in Section 3.2.

Next we applied logistic regression with a leave-one-out approach to determine the optimal weights of each of the two models. The weights produced by the algorithm were $Score = 0.23DF + 0.77EM$ where DF denotes the dataset-based metafeatures and EM denotes the embedding-based features.

The sequential approach. In this approach we first calculate the score of the model trained on embedding-based features and then add it as an additional feature to the set of dataset-based features. We then use the XGBoost algorithm to rank all possible algorithms.

4 EVALUATION

4.1 Experimental setup

We evaluated our approach on the well-known dataset published by [3], which contains the evaluation results of 179 classification algorithms on 119 datasets[1]. The list of datasets covers the UCI database in its entirety (as of March 2013, excluding some large-scale problems) in addition to some real-world problems (please see [3] for details). For each dataset, all applicable algorithms were applied and evaluated using the accuracy metric. The large scale of the experiments and the diversity of both datasets algorithms ensure that the results were free from collection bias.
We used the following settings throughout out evaluation:

- All the dataset-algorithm run results are taken from [3], who conducted all experiments and released the full results.
- For all purposes of training and evaluation, we used a leave-one-out approach: for each evaluated dataset d_i, we trained the ranking classifier using meta-features from $d_j \in D$ where $i \neq j$.
- We used *relative maximum accuracy* (RMA), which is the percentage of the maximal accuracy obtained by *any* algorithm on the dataset, as the evaluation measure for the performance of the classification algorithms on the various datasets. The accuracy measure is calculated using the formula:
  $Accuracy = \frac{TP+TN}{TP+FP+TN+FN}$
  where TP, FP, TN and FN are the number of true-positive, false-positive, true-negative and false-negative samples respectively.
- Since many of the 179 algorithms evaluated in [3] are different implementations of the same algorithm (Random Forest, for example, has eight variations), we group the algorithms into 17 "families". AutoDi therefore produces a ranked list of families rather than algorithms.
- When comparing the results of different algorithm families on a given dataset, we use the highest performance recorded by any member of the family. This decision sets a higher bar for our proposed hybrid method, as it becomes more difficult to demonstrate an improvement over the baselines.
- In addition to comparing the results of our hybrid approach to the performance of each of its components, we also required a baseline. Since random selection is not adequate, we instead chose the algorithm with the *highest overall average performance* – the Random Forest algorithm [6]. Because we use the results of the best performing family-member for each dataset, the relative maximal accuracy of the Random Forest algorithm in our experiments is 96.4% instead of the 94.1% reported in [3].

[1] originally [3] presented 121 datasets, but in our paper 2 datasets were excluded from the evaluation due to their format and readability problems

4.2 Evaluation results

Table 1 and Figure 1 show the results of our evaluation. In addition to the Random Forest baseline and the models trained separately on the two feature sets presented in sections 3.1 and 3.2, we evaluated two versions of AutoDi, as presented in Section 3.3. The results show that the Random Forest baseline fared worst both with respect to overall performance and consistency, as shown by its standard deviation. We can also conclude that the embedding-based features outperform the dataset-based metafeatures by these two criteria. Finally, the results illustrate the merits of our hybrid approach, which obtained the highest performance overall and the smallest standard deviation. The consistency of our approach is illustrated in Figure 1, where the hybrid version of AutoDi very rarely demonstrates the occasional drop in performance observed both by of its feature sets when they are used separately.

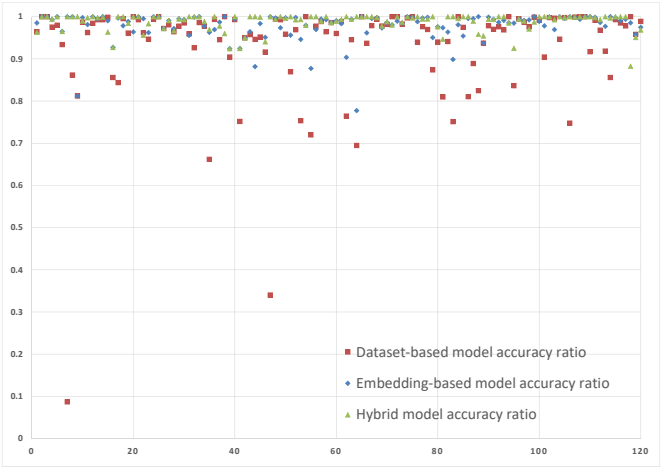

Figure 1: A scatter diagram showing the relative accuracy of each of the 119 datasets.

4.3 Results Analysis

We analyzed the results produced by the various models and reached the following conclusions:
The hybrid model is consistently better. Not only did AutoDi perform best with an RMA of 98.8%, it achieved perfect RAM on 54 of the 119 analyzed datasets. As shown in Table 2, this value is significantly higher than the other evaluated approaches.
The two feature sets complement each other. To better understand the reasons for the hybrid model's superior performance we

Table 1: The performance of the evaluated approaches. We use the relative maximal accuracy (RMA) measure.

Measure	Random Forest	Dataset-based	Embedding-based	AutoDi weighted averaging	AutoDi sequential
Average RMA	96.6	96.9	97.84	97.88	98.79
Stdev	6	3	4	3	2

Table 2: The number of datasets for each approach for which the optimal algorithm has been recommended.

Method Name	Number of Datasets with Top Performance
Random Forest	25 (21%)
Dataset-based	15 (12.6%)
Embedding-based	22 (18.4%)
AutoDi-sequential	**54 (49.5%)**

Table 3: The top-5 datasets for which the dataset-based and embedding-based models differed most in performance. Values are in absolute accuracy. Positive values indicate better performance by the latter model.

Dataset Name	Perf. Diff.	Suggested Cause
audiology-std	0.91	Rare dataset type in our training set + small number of instances
image-segmentation	0.66	Rare dataset type in our training set
flags	0.30	Rare dataset type in our training set
teaching	0.25	Small number of instances
libras	0.19	Small number of instances

compared the performance of the two feature sets – dataset-based and embedding-based – on the various datasets. Table 3 presents the five datasets with the largest difference in RMA. Our analysis shows that the embedding-based model is better in cases where the dataset is rarely used (i.e. not many similar cases in the training set) and/or has features that make learning difficult (i.e. a small number of instances or features). We conclude that in such cases deriving insights from previously published academic work is more useful than analyzing the dataset characteristics.

Next we compared the performance of the sequential hybrid model to that of the embedding-based model. Table 4 presents the five datasets with the largest difference in RMA performance. Our analysis shows that the hybrid approach performs better in cases where the description of the dataset lacks details or where the dataset is rarely used in academic research. Additional cases are where the data itself is highly imbalanced (the number of instances from the different classes differ greatly). We conclude that while the embedding-based approach is very effective, the hybrid model is capable of identifying scenarios in which particular characteristics of the data – size, imbalance etc. – need to impact the selection.

5 CONCLUSION AND NEXT STEPS

In this study we have presented AutoDi, an efficient framework for the model selection. By using a hybrid method that uses both dataset-based features and insights derived from a large corpus of academic publications, we were able to achieve a significant improvement in results over a large number of datasets.

For future work, we plan to explore recommending both the algorithm and its hyperparameter setting (as done in [4], [12]) as well as using our approach as a preliminary step for more computationally-intensive solutions such as AutoWeka [13].

Table 4: The top-5 datasets for which the hybrid model and the embedding-based model differed most in performance. Values are in absolute accuracy. Positive values indicate better performance by the former model.

Dataset Name	Perf. Diff.	Suggested Cause
monks-3	0.22	Problem description lacking
balloons	0.19	Small dataset, rarely used in academic papers
lung-cancer	0.12	Highly imbalanced, rarely used in academic papers
hill-valley	0.12	Unique problem, rarely used in academic papers
wine	-0.1	Very popular in academic papers

ACKNOWLEDGMENTS

This work has been supported in part by the Defense Advanced Research Projects Agency (DARPA) Data-Driven Discovery of Models (D3M) Program.

REFERENCES

[1] Pavel Brazdil, Christophe Giraud Carrier, Carlos Soares, and Ricardo Vilalta. 2008. *Metalearning: Applications to data mining.* Springer Science & Business Media.
[2] Pavel B Brazdil, Carlos Soares, and Joaquim Pinto Da Costa. 2003. Ranking learning algorithms: Using IBL and meta-learning on accuracy and time results. *Machine Learning* 50, 3 (2003), 251–277.
[3] Manuel Fernández-Delgado, Eva Cernadas, Senén Barro, and Dinani Amorim. 2014. Do we need hundreds of classifiers to solve real world classification problems. *J. Mach. Learn. Res* 15, 1 (2014), 3133–3181.
[4] Matthias Feurer, Aaron Klein, Katharina Eggensperger, Jost Springenberg, Manuel Blum, and Frank Hutter. 2015. Efficient and robust automated machine learning. In *Advances in Neural Information Processing Systems*. 2962–2970.
[5] Gilad Katz, Eui Chul Richard Shin, and Dawn Song. 2016. Explorekit: Automatic feature generation and selection. In *Data Mining (ICDM), 2016 IEEE 16th International Conference on*. IEEE, 979–984.
[6] Andy Liaw, Matthew Wiener, et al. 2002. Classification and regression by randomForest. *R news* 2, 3 (2002), 18–22.
[7] Tomas Mikolov, Kai Chen, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).
[8] Yonghong Peng, Peter A Flach, Carlos Soares, and Pavel Brazdil. 2002. Improved dataset characterisation for meta-learning. In *International Conference on Discovery Science*. Springer, 141–152.
[9] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 1532–1543.
[10] Bernhard Pfahringer, Hilan Bensusan, and Christophe G Giraud-Carrier. 2000. Meta-Learning by Landmarking Various Learning Algorithms.. In *ICML*. 743–750.
[11] Fábio Pinto, Carlos Soares, and João Mendes-Moreira. 2016. Towards automatic generation of metafeatures. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 215–226.
[12] Thomas Swearingen, Will Drevo, Bennett Cyphers, Alfredo Cuesta-Infante, Arun Ross, and Kalyan Veeramachaneni. 2017. ATM: A distributed, collaborative, scalable system for automated machine learning. In *IEEE Conference on Big Data*.
[13] Chris Thornton, Frank Hutter, Holger H Hoos, and Kevin Leyton-Brown. 2013. Auto-WEKA: Combined selection and hyperparameter optimization of classification algorithms. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 847–855.