Towards Robust Cluster-Based Hyperparameter Optimization*

Leonardo Izaú[1], Mariana Fortes[1], Vitor Ribeiro[2], Celso Marques[1],Carla Oliveira[1], Eduardo Bezerra[1], Fabio Porto[2], Rebecca Salles[1], Eduardo Ogasawara[1]

[1]Federal Center for Technological Education of Rio de Janeiro (CEFET/RJ)

[2]National Laboratory for Scientific Computing (LNCC)

{leoizau, mfortes}@eic.cefet-rj.br, victorr@posgrad.lncc.br

celso.silva@cefet-rj.br,carla.oliveira@ibge.gov.br,ebezerra@cefet-rj.br

fporto@lncc.br, rebeccapsalles@acm.org, eogasawara@ieee.org

Abstract. *Hyperparameter optimization is a fundamental step in machine learning pipelines since it can influence the predictive performance of the resulting models. However, the setup generally selected by classical hyperparameter optimization based on minimizing an objective function may not be robust to overfitting. This work proposes CHyper, a novel clustering-based approach to hyperparameter selection. CHyper derives a candidate cluster of close or similar hyperparameters with low prediction errors in the validation dataset. Hyperparameters chosen are likely to produce models that generalize the inherent behavior of the data. CHyper was evaluated with two different clustering techniques, namely k-means and spectral clustering, in the context of time series prediction of annual fertilizer consumption. Complementary to minimizing an objective function, cluster-based hyperparameter selection achieved robustness to negative overfitting effects and contributed to lowering a generalization error.*

1. Introduction

The recent rise in machine learning stems from the need for applications to efficiently process large volumes of data. Among the various machine learning applications, the ones provided by supervised learning models stand out. In the context of predictive models, several factors influence the predictive performance of the algorithm even before training starts, such as the choice of methods for data preprocessing and hyperparameters [García et al., 2014]. Hyperparameters are values that make up the initial configuration of the learning algorithm. We define a *hyperparameter setting* for a learning algorithm $\mathcal{A}$ as a tuple of assignments to each hyperparameter in $\mathcal{A}$.

For the algorithm to make predictions with greater accuracy, it is necessary to optimize the hyperparameters [Liu et al., 2021]. The Grid Search approach is commonly adopted to explore a broad range of hyperparameter settings. It consists of repeatedly training the learning algorithm with different possible hyperparameter settings combinations. At the end of the process, the hyperparameter setting that resulted in the lowest prediction errors (measured in a separate validation set) is chosen [Khalid and Javaid, 2020]. Such optimized hyperparameter settings can then be used to fit the learning model [Ran

*The authors thank CNPq, CAPES (finance code 001), and FAPERJ for partially sponsoring this research.

and Hu, 2017]. Since the number of hyperparameters can be high, adjusting each one and analyzing model behavior is challenging [Liu et al., 2021]. Although relevant, this challenge is outside the scope of this paper.

However, another relevant challenge occurs when the optimized hyperparameter, corresponding, for example, to the lowest global prediction error in machine learning, results in low accuracy in the test dataset. It is commonly related to the problem of data overfitting. Overfitting occurs when the fitted model is too dependent on the training dataset. One consequence is the lack of ability for the fitted model to generalize to unseen data observations [Sarwar Murshed et al., 2022]. When faced with the problem of overfitting, one must search for robust solutions in order to select optimized hyperparameters. In this case, choosing the hyperparameters that help the model reach the global minimum prediction error may not be enough to reach robustness.

Consider, for example, that a Grid Search for hyperparameters establishes a search space hyperparameters (*i.e.*, a set of hyperparameters). Each hyperparameter setting has an associated prediction error. Figure 1.a schematically illustrates the problem. Consider that each dot (red or blue) is a hyperparameter setting for a given learning algorithm in this picture. The set of considered hyperparameter settings is split into two clusters. Also, the hyperparameter setting corresponding to the minimum global prediction error (double-lined, green) in the validation dataset is clustered with similar hyperparameter settings. Here, closeness or similarity between hyperparameter settings refers to low vector distances. All hyperparameter settings close to the one with minimum global prediction error result in high prediction errors in the validation dataset. In this scenario, the hyperparameters with minimum global prediction error might be particularly suited for the training set by luck.

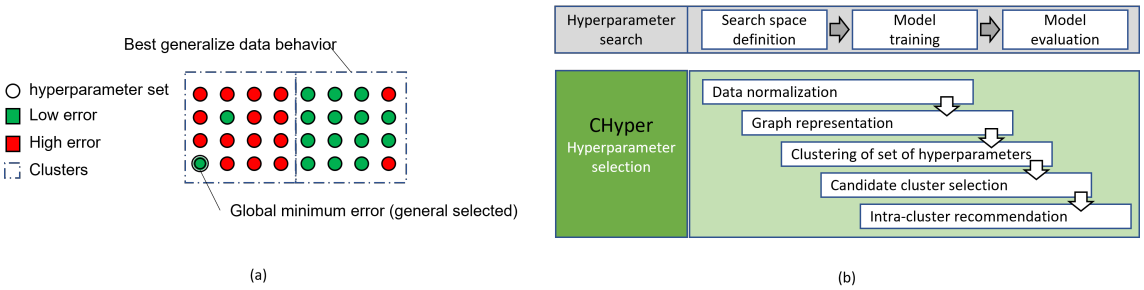

Figure 1. Hyperparameter selection problem (a), CHyper approach (b)

Considering this information, it seems more interesting to identify a cluster of similar hyperparameter settings that result in low prediction errors in the validation dataset. Thus, hyperparameter settings chosen from this cluster are more likely to allow the learned model to generalize better the inherent behavior of the data than the one with the minimum global error. In this context, the main objective of this work is to evaluate the hypothesis that selecting one among a group of close (similar) hyperparameter settings that generally result in low prediction errors in the validation dataset is better than selecting one that directly minimizes errors agnostically to its neighborhood. The proposed optimization approach, which we call `CHyper`, thus focuses on the phase of hyperparameter selection. First, it clusters the set of available hyperparameter settings.

Then, it considers the neighborhood of each candidate solution to drive the selection. The proposed solution, inspired by community detection techniques, aims to find communities of the most connected hyperparameter settings regarding configuration and prediction performance [Fortunato, 2010]. The intuition is that this approach achieves robustness in the hyperparameter setting and mitigates the negative effects of overfitting.

The related work was driven to search for papers that addressed hyperparameter optimization. The articles were evaluated by the degree of relevance to the proposed work. Selected articles can be divided into three groups. The first group involves works that propose hyperparameter optimization methods. For example, Yu and Kang [2019] proposes an optimization method based on clustering, but only for data partitioning, in which the data sample for training contains only examples of the target class. The second group applies Graph Theory to aid prediction. Zhang et al. [2019] proposes a composition of neural networks and graphs hyper-networks that generate weights for any architecture through its computational graphic representation. Finally, the third group presents articles that use clustering to aid prediction, such as Li and Huang [2021], which uses k-means to optimize the settings of an RBF Neural Network to predict stock values in the financial market. From related work, it was observed that the selection of hyperparameters has not yet been extensively studied. Therefore there is an opportunity for research in the area. Moreover, no direct work exploited clustering techniques for selecting hyperparameters.

Besides this introduction, this work is divided into three sections. Section 2 details the CHyper method. Section 3 presents the experimental setup and discusses the obtained results. Finally, Section 4 describes the main contributions of the proposal and possible future developments.

2. CHyper

This section presents the proposed approach for hyperparameter selection: CHyper. The general idea of CHyper is to find a candidate cluster of close hyperparameter settings that generally result in low prediction errors in the validation dataset. Instead of selecting hyperparameter settings that directly minimize error, CHyper recommends hyperparameters that are more likely to generalize the inherent data behavior better. The general steps of our approach are illustrated in Figure 1.b.

Consider a training-validation dataset scenario. A particular hyperparameter search process generally encompasses the phases of (i) hyperparameter search space definition, (ii) training of predictive models for each combination of mapped hyperparameters, and (iii) evaluation of the prediction accuracy achieved by each trained model based on a validation dataset. This process can produce a matrix $M_{n\times(m+1)}$, where m is the number of hyperparameters of interest, and n is the number of trained models. The matrix M includes an additional column corresponding to the prediction error value.

Consider a matrix M as input. The proposed CHyper hyperparameter selection approach outputs a hyperparameter setting that contributes to low prediction errors and provides model generalization. The general steps comprising CHyper are presented in Figure 1.b. In the first step, the $m + 1$ columns of matrix M are normalized using a particular normalization method, such as *z-score* [Ogasawara et al., 2009]. This step is important to ensure all scales are normalized and comparable, avoiding biased distances between hyperparameter vectors. The result is given in matrix M'.

The second step produces a complete graph $G = (V(G), E(G))$, where each row vector of M' corresponds to a vertex ($|V(G)| = n$). G is then computationally represented by a distance matrix D or the similarity matrix S. The matrix D (or S) is input to the third step consisting of applying a clustering method for deriving clusters of closer (similar) hyperparameters in configuration. They are expected to produce models with similar prediction performance. An adequate distance (similarity) function must be selected as a criterion for the adopted clustering method for the second and third steps. Examples include the Euclidean distance and Gaussian RBF. Methods such as k-means and spectral clustering also require choosing the number k of clusters in the output. At the end of the clustering process, a set of k clusters of hyperparameters is returned.

In the fourth step, the k clusters are compared to select a candidate cluster of close hyperparameters that generally result in low prediction error metrics ($eval$). Note that the selected candidate cluster may not contain the set of hyperparameters that minimizes prediction errors in M. Instead, we focus on clusters that contain hyperparameters with similar configurations that consistently contribute to relatively good prediction performances. The possible techniques to select the candidate cluster are (i) error-based and (ii) centrality-based. The error-based selects the cluster whose components summarize the lowest average of prediction errors. The centrality-based calculates a centrality measure for each vertex in a cluster-induced subgraph. The cluster with the most central vertex contributing to the lowest prediction error is chosen.

The last step corresponds to the final recommendation of a hyperparameter setting chosen from the selected candidate cluster. For that choice, different criteria can be adopted. For example, the recommended hyperparameter setting can also be (i) Error-based and (ii) Centrality-based. The error-based minimizes intra-cluster prediction errors. The centrality-based corresponds to the most central vertex in the cluster-induced subgraph.

3. Proof-of-concept experimental evaluation

The proposed method was experimentally evaluated in the context of time series prediction. For that, it was derived a time series dataset from public data available at the International Fertilizer Association (IFA)[1]. It contains data on the annual consumption of three fertilizers (K_2O, N, P_2O_5) in each of the world's top ten consumers (Brazil, Canada, China, France, India, Indonesia, Pakistan, Russia, Turkey, United States). The dataset consists of 30 time series. Each one contains 58 observations from 1961 to 2018. Observations for the period 1961-2002 are used for training, observations for 2003-2010 are used for validation, and the observations for 2011-2018 are used for testing.

3.1. Hyperparameter search

This work adopted a machine learning (ML) model to predict each fertilizer consumption time series contained in the adopted dataset. The adopted ML model was the Multilayer Perceptron Neural Network (MLP), one of the most common neural network architectures. It takes as main hyperparameters the number of neurons in the hidden layer ($neurons = [3, 8]$), the number of input entries ($inputsize = [3, 8]$), and the parameter for weight decay ($decay = [0.1, 1.0]$).

[1] http://www.fertilizer.org

Based on the defined search space, 396 possible hyperparameter settings are generated, resulting in a matrix $M_{n \times m}$, $m = 3$ and $n = 396$. Each hyperparameter setting (row) is given as input for training a corresponding MLP model. This process is conducted based on the training dataset of each time series previously normalized by Min-max [Ogasawara et al., 2009].

The prediction performance of each model is evaluated over the validation dataset based on the symmetric mean absolute percentage error (sMAPE). Lower sums of prediction errors over cross-validation indicate higher prediction performance and more accurate predictions. Finally, the evaluated errors for models trained with each hyperparameter are included in the last M ($m = 4$). By the end of the process, 90 M-like matrices are produced (30 time series $\times$ 3 fold setups in cross-validation) and given as input to the evaluation of CHyper.

3.2. Experimental setup

This section presents the experimental setup adopted to conduct the necessary steps for CHyper hyperparameter selection based on M. Particularly, data normalization is done by *z-score* [Ogasawara et al., 2009]. The Euclidean distance and the Gaussian RBF are adopted as vector distance measures and similarity functions (kernel), respectively. Heuristics automatically determine suitable values for the parameter sigma of the RBF function [Karatzoglou et al., 2004]. Eigenvector centrality measures calculate vertex centrality in cluster-induced subgraphs.

Both k-means and Spectral clustering perform clustering of the set of hyperparameters. The number of clustering centroids is determined by the elbow method. In this case, the number of clusters is set to $k = 3$ after simulations of k in the discrete interval $2 \leq k \leq 10$. Finally, the choice of the candidate cluster of hyperparameter settings and the intra-cluster recommendation are either error-based or centrality-based (Section 2). Particularly, the matrix used by k-means is transformed into a similarity matrix $S' = (s'_{i,j})$, where $s'_{i,j} = (dist(x_i, x_j))^{-1}$, for running the centrality after clustering. In the case of Spectral clustering, the matrix used to obtain the centralities is precisely S.

The experimental results are compared with the traditional hyperparameter selection (*i.e.*, based on the lowest average prediction error in the training dataset). The proposed CHyper hyperparameter selection approach is currently available at GitHub[2].

3.3. Results and discussion

The first experimental evaluation target the selection of clusters, *i.e.*, error-based versus centrality cluster selection) for both clustering techniques. For spectral clustering, in 66% of cases, error-based was better. However, for k-means, 100% of the error-based was better. Due to that, the next experimental evaluation target error-based cluster selection.

The second experimental evaluation target the actual hyperparameter selection. Evaluated methods are referenced as CHyper(spec) and CHyper(k-means), which correspond to the hyperparameters recommended by CHyper using Spectral Clustering and k-means. Besides, Traditional corresponds to selecting the hyperparameters that minimize average prediction error in the training dataset.

[2]https://github.com/cefet-rj-dal/clusterhyper

The performance of methods was evaluated for all input datasets (90). Overall, CHyper outperformed the Traditional approach for most adopted datasets (51.1%). Spectral Clustering resulted in 16.7% of the best results, while k-means was responsible for 14.4% of them. In 20% of the cases, Spectral Clustering and k-means tied, making the same hyperparameter recommendation and outperforming the Traditional. These results indicate the potential of CHyper to overcome overfitting effects and improve hyperparameter selection and the process of time series prediction.

4. Conclusion

This work investigates alternatives to aid hyperparameter selection by proposing CHyper. It is a novel hyperparameter selection approach named CHyper. It recommends hyperparameters that have surrounded hyperparameters with low prediction errors in the test dataset. CHyper was competitive regarding prediction performances, recommending hyperparameters contributing to lower prediction errors in the test dataset for more than half of the time series. The paper opens room for deeper studies on hyperparameter selection. Additionally, the CHyper hyperparameter selection approach can be adopted by optimization methods currently available in the literature and is not limited to Grid Search.

References

Fortunato, S. (2010). Community detection in graphs. *Physics reports*, 486(3-5):75–174.

García, S., Luengo, J., and Herrera, F. (2014). *Data Preprocessing in Data Mining*. Springer.

Karatzoglou, A., Hornik, K., Smola, A., and Zeileis, A. (2004). kernlab - An S4 package for kernel methods in R. *Journal of Statistical Software*, 11:1–20.

Khalid, R. and Javaid, N. (2020). A survey on hyperparameters optimization algorithms of forecasting models in smart grid. *Sustainable Cities and Society*, 61.

Li, H. and Huang, S. (2021). Research on the Prediction Method of Stock Price Based on RBF Neural Network Optimization Algorithm. In *E3S Web of Conferences*, volume 235.

Liu, Y., Sun, Y., Xue, B., Zhang, M., Yen, G., and Tan, K. (2021). A Survey on Evolutionary Neural Architecture Search. *IEEE Transactions on Neural Networks and Learning Systems*.

Ogasawara, E., Murta, L., Zimbrão, G., and Mattoso, M. (2009). Neural networks cartridges for data mining on time series. In *IJCNN*, pages 2302–2309.

Ran, Z.-Y. and Hu, B.-G. (2017). Parameter identifiability in statistical machine learning: A review. *Neural Computation*, 29(5):1151–1203.

Sarwar Murshed, M., Murphy, C., Hou, D., Khan, N., Ananthanarayanan, G., and Hussain, F. (2022). Machine Learning at the Network Edge: A Survey. *ACM Computing Surveys*, 54(8).

Yu, J. and Kang, S. (2019). Clustering-based proxy measure for optimizing one-class classifiers. *Pattern Recognition Letters*, 117:37–44.

Zhang, C., Ren, M., and Urtasun, R. (2019). Graph hypernetworks for neural architecture search. In *ICLR, 2019*.