Mapped supervised learning

Guillermo Hernández[1], Angélica González Arrieta[1], Pablo Chamoso[1] and Juan M. Corchado[1,2,3]

[1]Grupo de Investigación BISITE, Departamento de Informática y Automática, Facultad de Ciencias, Universidad de Salamanca, Pl. Caídos, s/n, 37008 Salamanca, Spain

[2]Air Institute, IoT Digital Innovation Hub, 37188 Salamanca, Spain

[3]Department of Electronics, Information and Communication, Osaka Institute of Technology, 535-8585 Osaka, Japan

Abstract

Ensemble methods have become increasingly popular in machine learning due to their ability to improve model performance and generalizability. In this paper, we introduce the Mapped Supervised Learning (MSL) paradigm, an ensemble approach designed for mapping-based subproblem decomposition scenarios. MSL enables the application of supervised learning algorithms to subproblems defined by mappings to finite sets of integers, which has been shown to improve performance on some datasets, as confirmed by statistical tests. Moreover, MSL can enhance the explanatory power of resulting models, particularly in scenarios with multiple independent subproblems where conventional models may lack sufficient abstraction capacity. The MSL paradigm represents a valuable addition to the ensemble methods toolkit and holds promise for improving the accuracy and interpretability of machine learning models in a variety of applications.

Keywords

machine learning, supervised learning, ensemble learning, explainable machine learning

1. Introduction

Ensemble methods have emerged as a powerful approach to machine learning, offering improved performance, robustness, and generalizability compared to traditional single-model approaches. Ensemble methods work by combining the predictions of multiple models, typically through some form of voting or averaging, with the aim of reducing bias and variance and improving accuracy. While ensemble methods have been shown to be effective in many applications, they also come with their own set of challenges, such as increased computational complexity, higher memory requirements, and potential overfitting [1].

Ensemble methods are widely used in machine learning to improve the performance of predictive models by combining the outputs of multiple base models, usually fit with the same training method. Two main families of ensemble methods are commonly distinguished based on their underlying principles: averaging and boosting.

IVUS 2023

guillehg@usal.es (G. Hernández); angelica@usal.es (A. González Arrieta); chamoso@usal.es (P. Chamoso); corchado@usal.es (J. M. Corchado)

0000-0002-7481-5961 (G. Hernández); 0000-0002-4726-7103 (A. González Arrieta); 0000-0001-5109-3583 (P. Chamoso); 0000-0002-2829-1829 (J. M. Corchado)

Averaging methods aim to reduce the variance of a predictive model by building several base models independently and then combining their predictions. This approach reduces the risk of overfitting to the training data and improves the generalization performance of the model. A well-known example of averaging methods is bagging [2], which uses bootstrap samples to train multiple base models and combines their predictions by taking a simple average. Variations using other sampling methods exist as well [3]. Another example is the random forest algorithm, which builds an ensemble of decision trees by randomly selecting subsets of features and observations at each node split [4].

Boosting methods, on the other hand, aim to reduce the bias of a predictive model by building a sequence of base models that focus on the misclassified instances from the previous models. Boosting combines several weak models to produce a powerful ensemble, where each model contributes its expertise to the final prediction. A popular example of boosting is AdaBoost [5], which assigns higher weights to misclassified instances and trains a new base model on the updated weighted training set. Gradient tree boosting [6] is another example of boosting that uses gradient descent to minimize the residual error between the predicted and true values.

In addition to averaging and boosting, we propose a new ensemble method called the "mapping method". This method involves fitting a set of independent models with disjoint subsets of data using the same base learner. The mapping method decomposes complex problems into simpler sub-problems that can be solved independently, reducing the complexity of the problem and improving model accuracy. Our proposed method enables the use of different types of base learners and can be easily parallelized for efficient computation. We demonstrate the effectiveness of the mapping method on various machine learning applications in this paper.

The remainder of this paper is organized as follows: Section 2 provides a formal description of the proposed method. In Section 3, we present the results of a statistical evaluation of the method using multiple regression tasks for a dataset. Finally, in Section 4, we summarize our conclusions and discuss possible future directions for research.

2. Formal description

Supervised machine learning algorithms can be described, following [7], as an application $\mathcal{A}$ mapping a collection of instances of a set $\mathcal{X}$ with a label in a set $\mathcal{Y}$ into the a so called model, which is itself an application from $\mathcal{X}$ to $\mathcal{Y}$, chosen from a subset of applications $\mathcal{F}$ which are the candidate models. Formally:

$$\mathcal{A}\colon \bigcup_{m\in\mathbb{N}} (\mathcal{X}\times\mathcal{Y})^m \to \mathcal{F}\,. \tag{1}$$

Typically, these algorithms minimize a loss function, such as the quadratic loss $\mathcal{L}(f,(x,y)) = (f(x)-y)^2$, which is commonly used in one-dimensional real regression problems. However, the development of this work does not rely on the notion of loss, and the algorithms can be viewed solely as applications in the form of Equation 1.

To introduce the mapping ensemble approach, consider any surjetive mapping into a finite set of rational numbers, $\phi : \mathcal{X} \to \{1,\dots,n\}$. This decomposes the input space $\mathcal{X}$ into the n

disjoint subsets defined by the inverse application ϕ^{-1}, i.e., $\left\{\mathcal{X}_i = \phi^{-1}(i)\, \forall i \in 1, \ldots, n\right\}$ such that $\bigcup_{i=1}^{n} \mathcal{X}_i = \mathcal{X}$ and $\mathcal{X}_i \cap \mathcal{X}_j \neq \emptyset$ if and only if $i \neq j$.

The mapping ensemble approach is based on the previously described decomposition of the input space into disjoint subsets. This approach utilizes a set of independent models, all trained with the same base learner, with each model being trained on a different subset $\mathcal{X}_i$. Specifically, the mapping of an algorithm $\mathcal{A}$ using a map ϕ is an application $\mathcal{M}_{\mathcal{A},\phi}$ defined as

$$\mathcal{M}_{\mathcal{A},\phi}\left((x_i, y_i)\forall i \in 1 \ldots m\right)(x) = \mathcal{A}\left((x_i, y_i)\forall i \colon \phi(x_i) = \phi(x)\right)(x)\,. \tag{2}$$

It is important to note that the application defined in 2 follows the same form as the supervised learning algorithms described in 1. Therefore, the mapping ensemble approach can be seen as a supervised learning algorithm itself.

The choice of mapping function ϕ is a hyperparameter that can be optimized using standard procedures [8, 9]. Some natural options include using bijections for categorical attributes and mapping real-valued attributes to the index of subintervals of a partition of their domain after scaling transformations.

In the case of a classification problem, an alternative algorithm $\mathcal{B}$ can be used to construct ϕ. The resulting mapping ensemble can be represented as $\mathfrak{M}_{\mathcal{A},\mathcal{B}}$, defined as:

$$\mathfrak{M}_{\mathcal{A},\mathcal{B}}\left((x_i, y_i)\forall i \in 1 \ldots m\right) = \mathcal{M}_{\mathcal{A},\psi\circ\mathcal{B}(((x_i,y_i)\forall i\in 1\ldots m))}\left((x_i, y_i)\forall i \in 1 \ldots m\right)\,, \tag{3}$$

where ψ is an arbitrary bijection from the set of n classes to the first n integers.

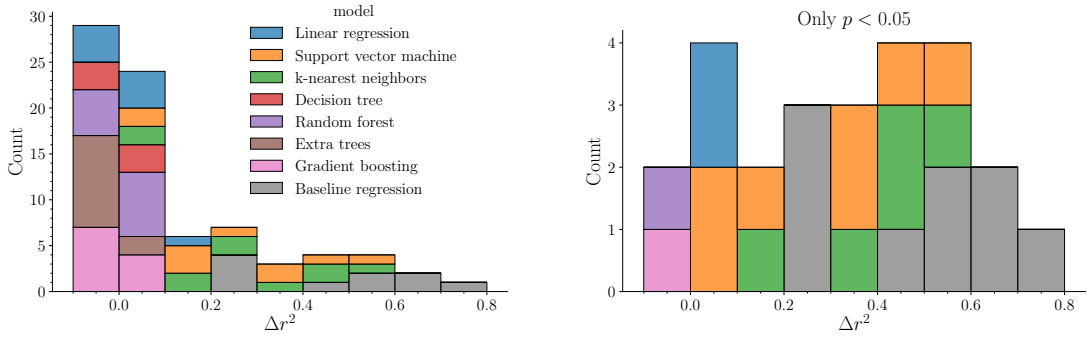
For regression problems, a similar strategy can be applied by replacing ψ with a partition as described above.

3. Evaluation

To show that mapped supervised ensemble is able to improve the results of some learning algorithms, we will apply it to a simple set of problems built using the UCI Wine Data Set [10], which contains chemical analysis results for wines grown in the same Italian region by three different cultivators. This dataset includes 13 attributes, one of which is the class label, and the remaining 12 are quantitative measurements of constituents found in the wines. While the dataset is commonly used for initial testing of new classification models, we will use it for a different purpose. Specifically, we will treat the class label as a categorical attribute and create 12 separate regression tasks using the remaining attributes but one to predict the value of the one left out. Note these tasks can also be regarded as imputation-building tasks.

We have employed a set of commonly used supervised learning methods from the scikit-learn library [11] to study these tasks. The methods considered include linear regressions, support vector machines, k-nearest neighbors, decision trees, random forests, extra randomized trees, gradient boosting, and baseline regression. We used the default hyperparameter values available in v1.2.2 for all the methods.

The mapping ϕ is defined as the projection of the first attribute (sometimes denoted as Π_1), which is here the categorical attribute. We have chosen this simple mapping to study the scenario where a simple categorical attribute suggests splitting the dataset. Although more



(a) All of the results

(b) Statistically significant results ($p < 0.05$)

Figure 1: Increase of the r^2 distribution in mapped regressors for the UCI Wine Data Set.

complicated mappings could be used, including the use of other machine learning methods, for the purpose of this work, we will focus on this simpler scenario.

The performance of the models will be evaluated using r^2 to enable comparison across tasks. To determine whether the differences between the original model and the mapped model are statistically significant, we will use the 5x2cv paired t test, which was proposed by Dietterich to address the limitations of traditional cross-validation or resampling tests [12]. This allows us to obtain measures of r^2 for both the original and mapped methods, and to calculate their difference Δr^2, along with a p-value that can be used to determine if such a difference is statistically significant.

Experiments where both the original model and the mapped model have an r^2 score less than 0.1 will be excluded from the analysis. This is because any model whose performance is not superior to the baseline model should be rejected in favor of the baseline model. Additionally, the failure to pass the statistical test could also be influenced by a lack of sufficient data. Therefore, we will report two types of analyses: one that includes all results and another that only includes statistically significant results.

Figure 1 displays the evaluation results, with Figure 1a presenting all the results (provided that at least one of the two r^2 scores is greater than 0.1, as explained earlier), and Figure 1b presenting only the statistically significant results ($p < 0.05$). As shown, the mapping is effective in improving the performance of simple models such as support vector machines and k-nearest neighbors, sometimes in a large amount in terms of r^2. These models have the added benefit of being easy to interpret and providing reasonable extrapolations, particularly in the case of linear models. Baseline models (which here always predict the mean value) also benefit from the mapping, demonstrating that when a clear option exists to split a dataset, the concept of a baseline model can be adapted accordingly.

However, tree-based models such as decision trees, extra randomized trees, and gradient boosting do not exhibit any improvement. This is unsurprising, as these algorithms naturally incorporate dataset splitting into their behavior. Nevertheless, these algorithms could be used as an alternative method to define a mapping, which we plan to explore in future work.

4. Conclusions

In this paper, we have introduced mapping supervised learning, an ensemble method with applications to supervised learning. Our proposed method was evaluated on multiple regression tasks using the UCI Wine Data Set and a simple mapping function defined by the original class label. Our findings indicate that the method can significantly improve the results of several regression methods, including support vector machines and k-nearest neighbors, as well as improving baseline models.

However, we have also observed that tree-based methods, which naturally incorporate the discovery of the simple mapping used here in their algorithms, do not significantly benefit from the application of the mapping ensemble technique. In future work, we plan to propose an alternative mapping construction technique that could improve the performance of tree-based models.

Moreover, we aim to further evaluate the supervised learning application of the mapping technique and explore its potential application to other paradigms of machine learning. Additionally, we intend to investigate the interpretability of the mapped models and how the mappings can be used to gain insights into the relationships between features and the target variable. We believe that the proposed mapping ensemble technique has significant potential for enhancing the performance and interpretability of supervised learning models.

References

[1] X. Dong, Z. Yu, W. Cao, Y. Shi, Q. Ma, A survey on ensemble learning, Frontiers of Computer Science 14 (2020) 241–258. doi:https://doi.org/10.1007/s11704-019-8208-z.

[2] L. Breiman, Bagging predictors, Machine learning 24 (1996) 123–140. doi:https://doi.org/10.1007/BF00058655.

[3] L. Breiman, Pasting small votes for classification in large databases and on-line, Machine learning 36 (1999) 85. doi:https://doi.org/10.1023/A:1007563306331.

[4] L. Breiman, Random forests, Machine learning 45 (2001) 5–32. doi:https://doi.org/10.1023/A:1010933404324.

[5] Y. Freund, R. E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, Journal of computer and system sciences 55 (1997) 119–139. doi:https://doi.org/10.1006/jcss.1997.1504.

[6] J. H. Friedman, Greedy function approximation: a gradient boosting machine, Annals of statistics (2001) 1189–1232. doi:10.1214/aos/1013203451.

[7] J. Berner, P. Grohs, G. Kutyniok, P. Petersen, The Modern Mathematics of Deep Learning, Cambridge University Press, 2022, p. 1–111. doi:10.1017/9781009025096.002.

[8] J. Bergstra, Y. Bengio, Random search for hyper-parameter optimization., Journal of machine learning research 13 (2012). doi:10.5555/2188385.2188395.

[9] L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh, A. Talwalkar, Hyperband: A novel bandit-based approach to hyperparameter optimization, The Journal of Machine Learning Research 18 (2017) 6765–6816. doi:10.5555/3122009.3242042.

[10] M. Lichman, UCI machine learning repository, 2013. URL: https://archive.ics.uci.edu/ml.

[11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al., Scikit-learn: Machine learning in python, the Journal of machine Learning research 12 (2011) 2825–2830. doi:10.5555/1953048.2078195.

[12] T. G. Dietterich, Approximate statistical tests for comparing supervised classification learning algorithms, Neural computation 10 (1998) 1895–1923. doi:10.1162/089976698300017197.