METAMIMIC: ANALYSIS OF HYPERPARAMETER TRANSFERABILITY FOR TABULAR DATA USING MIMIC-IV DATABASE

Mateusz Grzyb
Faculty of Mathematics and Information Science
Warsaw University of Technology
m.grzyb@student.mini.pw.edu.pl

Zuzanna Trafas
Faculty of Computing and Telecommunications
Poznan University of Technology
zuzanna.trafas@student.put.poznan.pl

Katarzyna Woźnica
Faculty of Mathematics and Information Science
Warsaw University of Technology
k.woznica@mini.pw.edu.pl

Przemysław Biecek
Faculty of Mathematics and Information Science
Warsaw University of Technology
przemyslaw.biecek@pw.edu.pl

ABSTRACT

The performance of boosting algorithms is highly susceptible to selecting appropriate hyperparameter values, which is computationally expensive and time-consuming. Transfer learning is one of the answers to the strong demand for more efficient methods of hyperparameter optimisation. It is proven that using model performance results for unrelated tasks allows faster tuning compared to traditional approaches, but few attempts have been made to answer what affects the transfer capability. To fill this gap, we created a repository of medical domain machine learning tasks based on the MIMIC-IV database and considered a few scenarios of different structural task resemblance, reflecting real-life use-cases. Results suggest that structural similarity enhances the transferability of hyperparameters, which may, from a practical application perspective, mean it is beneficial to store tuning history when dealing with tasks of similar definition. The code needed to reproduce the study can be found in this repository: https://github.com/ModelOriented/metaMIMIC.

1 Introduction

The use of analogy is one of the elementary methods of problem-solving. It is a process of information transfer from solutions to related problems. For humans, it is natural to base on previously learned skills and experience when learning new abilities, but it is not so evident for machines. In the field of machine learning (ML), the approach of knowledge transfer is represented by the subject of transfer learning (TL). TL focuses on reusing knowledge gained while solving past problems for the faster creation of better performing models.

TL is a widely used technique in deep learning, an approach especially suited for tasks like computer vision and speech recognition [5]. For example, knowledge gained while training a neural network to recognise cats in pictures could be applied to a dog recognition task. However, neural networks are usually not the best choice when working with tabular data, where boosting models are better suited. Importantly, realising their full potential is dependent on the appropriate choice of a whole range of internal settings called hyperparameters. It is a complex task with high computational cost, and there is no single methodology that would work best in every scenario.

Our study aims to explore hyperparameter transfer possibilities for boosting algorithms, from which the process of model tuning could benefit significantly. For this purpose, we used the MIMIC-IV [6] medical database to prepare a few scenarios of different structural task resemblance. We conducted computational experiments using the XGBoost algorithm [2] and a parameter grid derived from the MementoML study [8] and analysed them to assess the possibility of the mentioned hyperparameter transfer. In addition, using the results obtained, we simulated the model tuning process using hyperparameter transfer to assess the potential speed improvement.

2 Data preparation

In this section, we introduce the MIMIC-IV database and present the process of preparing the data for planned experiments. The choice of this database was dictated by the large amount of diverse data which made it possible to construct many prediction tasks on its basis.

2.1 MIMIC-IV database

MIMIC-IV (Medical Information Mart for Intensive Care) is a large, freely available database comprising de-identified health-related data from patients who were admitted to the intensive care unit (ICU) of the Beth Israel Deaconess Medical Center. It contains data of over 380,000 patients admitted to the ICU in years the 2008-2019. In our analysis, we used three MIMIC-IV modules – `core`, `hosp`, and `icu`:

- `core` module contains patient tracking data. We used it to select a list of unique patients along with some demographics (`patients` and `admissions` tables).
- `hosp` module provides all data acquired from the hospital wide electronic health record. We used it to get billed ICD-9 / ICD-10 diagnoses for hospitalizations (`diagnoses_icd` table) and laboratory measurements sourced from patient derived specimens (`labevents` and `d_labevents` table).
- `icu` module contains information collected from the clinical information system used within the ICU. We used it to get charted events occurring during the ICU stay (`chartevents` and `d_items` tables).

2.2 Cohort selection

Following previous studies [7, 9, 10], we defined the following patient inclusion criteria:

- We consider only the first admission of every patient to preserve independence of all observations.
- Every patient must be at least 15 years old at the time of their stay (lower ages are censored with a value 0 in the database)
- We include patients with at least one chartevent, one labevent, and one diagnosis corresponding to the selected admission.
- The hospital stay length must be less than 60 days.

In total, 34925 unique patients met all the above conditions.

2.3 Target selection

Diagnosis codes The International Classification of Diseases (ICD) is a globally used diagnostic tool for epidemiology, health management, and clinical purposes. There are over 69,000 unique diagnosis codes in ICD-10 and roughly 14,000 diagnosis codes in ICD-9. In `diagnoses_icd` table we can find a record of all diagnoses a patient was billed for during their hospital stay.

Selection criteria We examined 50 most commonly appearing conditions and hand-picked groups of diseases that have a representation in both ICD-9 and ICD-10 codes (Table 1). It resulted in 12 targets for binary classification. We also considered whether the targets selected can be successfully predicted with the data available in the MIMIC-IV database (at least 0.7 mean 4-CV ROC AUC after tuning).

Table 1: Selected targets with corresponding ICD codes and frequency in the considered cohort.

Condition	ICD-9	ICD-10	Frequency
Hypertensive diseases	401-405	I10-I16	59.8%
Disorders of lipoid metabolism	272	E78	40.3%
Anemia	280-285	D60-D64	35.9%
Ischematic heart disease	410-414	I20-I25	32.8%
Diabetes	249-250	E08-E13	25.3%
Chronic lower respiratory diseases	466, 490-496	J40-J47	19.5%
Heart failure	428	I50	19.4%
Hypotension	458	I95	14.5%
Purpura and other hemorrhagic conditions	287	D69	11.9%
Atrial fibrillation and flutter	427.3	I48	10.5%
Overweight, obesity and other hyperalimentation	278	E65-E68	10.5%
Alcohol dependence	303	F10	7.7%

Target correlation To assess target correlation, we used Yule's Q, which is more robust than Pearson's r when dealing with binary data. Mostly, we observe low and medium positive coefficient values between targets. The only exception is alcohol dependence, which has a negative correlation with most of the other diagnoses. Exact Yule's Q values can be found in Appendix A (Figure 7).

2.4 Feature selection and extraction

We selected 58 features from the `chartevents` and `labevents` tables, hand-picking ones with the lowest number of missing values. We ignored the features related to the settings of hospital apparatus and categorical variables, for which the number of unique values would result in a high dimensional and sparse feature space. Moreover, we included age and gender from the `patients` table.

For most of the patients, 56 of these features were recorded several times. Therefore, we extracted the minimum, average and maximum values. For weight and height, we recorded only the first measurement. In total, this resulted in 172 variables. The list of selected predictors with more details can be found in Appendix B (Table 3).

3 Methodology

In this section, we describe the methodology of the conducted experiments. The main idea was to create multiple ML tasks of varying similarity based on the previously described data and then use them to evaluate a predefined grid of XGBoost hyperparameterss. Missing values were imputed with a mean of all observations for each task independently to avoid data leakage. Model fitness was assessed with a mean ROC AUC score from 4 cross-validation folds, which allowed us to create multiple rankings of considered hyperparameter sets.

The code needed to reproduce the study can be found in this repository: https://github.com/ModelOriented/metaMIMIC.

3.1 Hyperparameter grid

We used the grid from the MementoML study. It comprises 1000 sets of 8 different XGBoost hyperparameters sampled independently from predefined distributions. Considered hyperparameters and the distributions they were sampled from are presented in Table 2.

Table 2: Hyperparameters and their underlying distributions. U stands for a random variable sampled from a uniform distribution with corresponding lower and upper bounds. Booster can be either "gblinear" or "gbtree".

Hyperparameter	Type	Lower	Upper	Distribution
n_estimators	integer	1	1000	U
learning_rate	float	0.031	1	2^U
booster	discrete	-	-	U
subsample	float	0.5	1	U
max_depth	integer	6	15	U
min_child_weight	float	1	8	2^U
colsample_bytree	float	0.2	1	U
colsample_bylevel	float	0.2	1	U

Incorporating the same hyperparameter grid that was also used in the MementoML study allowed us to compare our results with ones obtained on 22 ML tasks from OpenML repository [12].

3.2 Experiment descriptions

To test when hyperparameter transfer can be used, we considered task pairs of different similarity levels. We started with tasks based on the same data and differing only in targets. Then, we loosened this conditions by introducing divergence in observations and variables used. The goal of this operation was to analyse the influence of task similarity on hyperparameter transfer possibilities.

The process of task creation process always consists of three choices – which predictors to use, which observations to consider and which target to predict. The number of options in each decision depends on the experiment and reflects the possible dissimilarity between the tasks. We visualised this process through decision making diagrams.

3.2.1 Experiment 1

The first experiment reflects the situation where we predict different targets considering the same observations and using the same variables. Therefore, the only real choice to make is to select which target to predict (Figure 1).

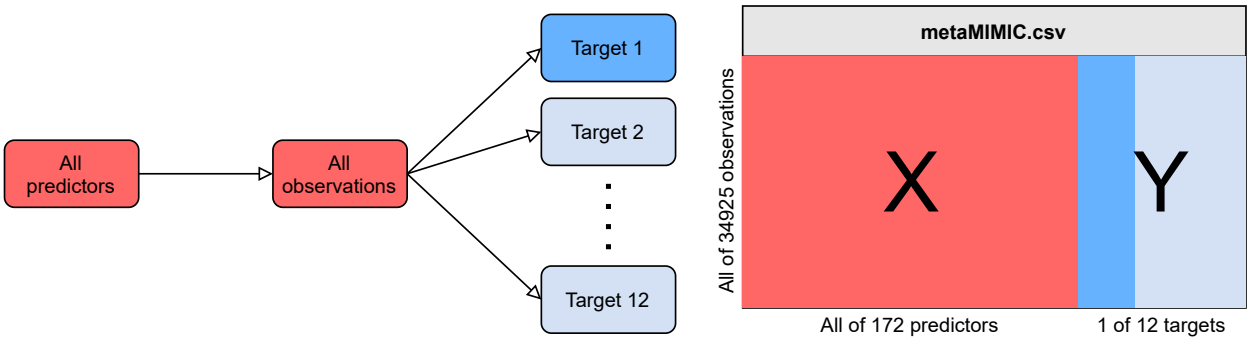

Figure 1: Decision making diagram of task creation in Experiment 1

Use case The above could be the case in the following scenario. First, consider a data scientist working at a hospital who is charged with preparing ML models for the prediction of multiple health conditions. They are provided with historical data of the hospital's patients, which comprises basic diagnostic tests and diagnoses. The only difference between the tasks is the diagnosis we want to predict. Therefore, TL could speed up the whole process and save resources.

3.2.2 Experiment 2

The second experiment reflects the situation where we predict different targets considering different observations but using the same variables. Divergence in considered observations is realised through a choice between two halves of observations, split in the same way for all targets (Figure 2). The data was sorted using all 12 targets to ensure a balance between these subsets.

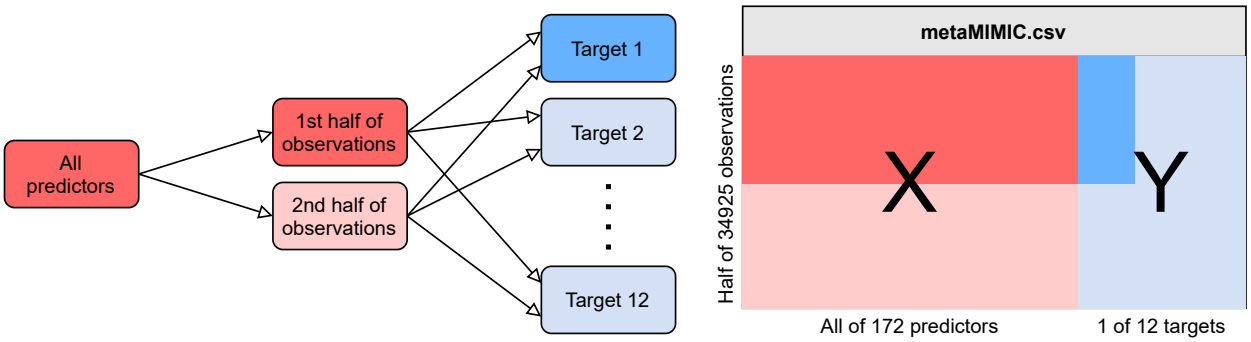

Figure 2: Decision making diagram of task creation in Experiment 2

Use case Consider the scenario from Experiment 1. Some time passes, and the hospital gathers data of many new patients. The hospital's management wants their data scientist to update all of the models. Again, if TL would be possible for datasets with different observations, it could shorten the process of tuning significantly.

3.2.3 Experiment 3

The third experiment reflects the situation where we predict different targets considering different observations and using different variables. Divergence in variables used comes not only from the different number of predictors but also from the fact that we used different sets of predictors for each task (Figure 3). Predictor set choice was realised through selecting top n variables with the highest permutation variable importance value [1, 4], calculated using XGBoost model with default settings.

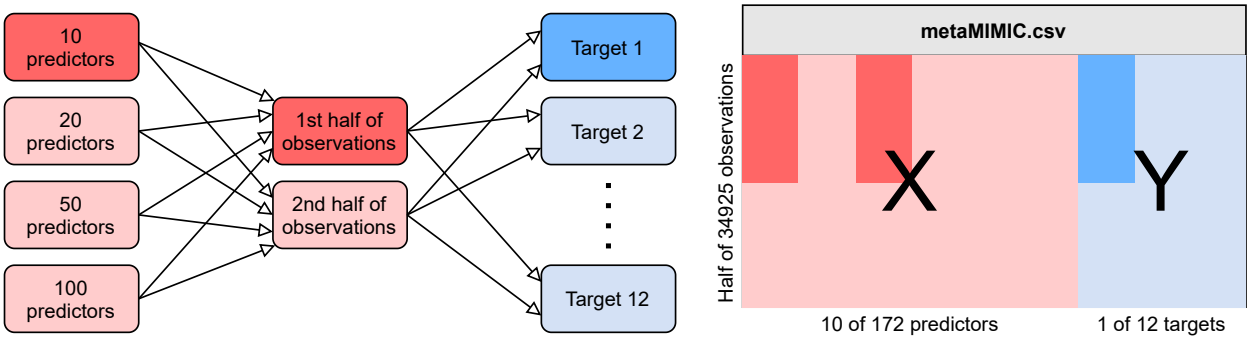

Figure 3: Decision making diagram of task creation in Experiment 3

Use case Let us follow with the scenario from the previous experiments. Our data scientist analyses the created models and realises that the dimensionality of the problem can be reduced. He wonders whether such feature engineering would make it necessary to tune the models again. It could be avoided if TL would still be possible, and therefore save computation time.

4 Results

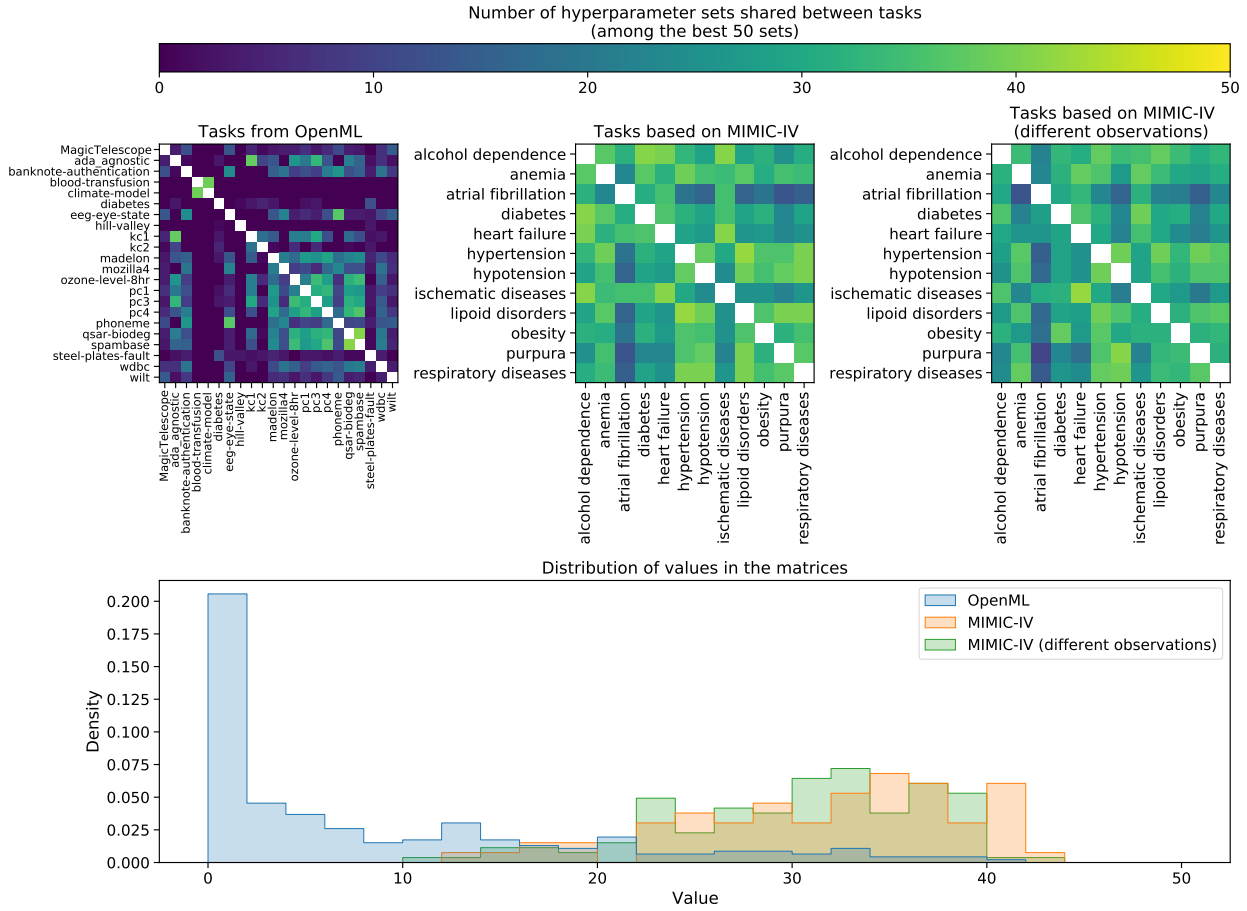

Figure 4: Numbers of the best 50 hyperparameter sets (regarding to the mean 4-CV ROC AUC measure) shared between tasks from OpenML, Experiment 1 and Experiment 2. An individual cell of the matrix corresponds to the number of hyperparameter sets shared between a given pair of tasks. Histograms summarise the distribution of the values of each matrix. White colour on the diagonal means that the value is not considered.

To analyse the results obtained from the first two experiments, we decided to look at the beginning of the ranking of the hyperparameter sets, decreasing in terms of the mean 4-CV ROC AUC, because naturally, these are the sets of highest interest when tuning a model. For this purpose, we examined how many of the best 5% hyperparameter sets (for each task individually) are shared between tasks from different sources (Figure 4).

Fluctuations in mean 4-CV ROC AUC may cause subtle permutations of rankings (noise). We decided on a threshold of 5% because it minimises the impact of this on the actual results. Nevertheless, choosing another threshold value from a reasonable range of 20-100 results in analogous relationships between the distributions of values in the matrices. In addition, this fact is also reflected in the mean of Spearman rank correlation coefficients calculated for individual pairs of full rankings (0.165 ± 0.469 for OpenML, 0.885 ± 0.072 for MIMIC-IV, and 0.849 ± 0.078 for MIMIC-IV (different observations)).

The comparison of the distributions of values in the presented matrices shows that the number of shared best hyperparameter sets is significantly higher for MIMIC-IV-based problems (representing a scenario of identical problem definitions) than for OpenML-based problems (representing a scenario of unrelated problems). In addition, with the rightmost matrix, it is apparent that considering disjoint subsets of observations (which is often closer to actual use-cases) results in only a slight decrease in the average number of shared hyperparameter sets. These results suggest that it is possible to make more effective use of hyperparameter transfer between similarly structured tasks.

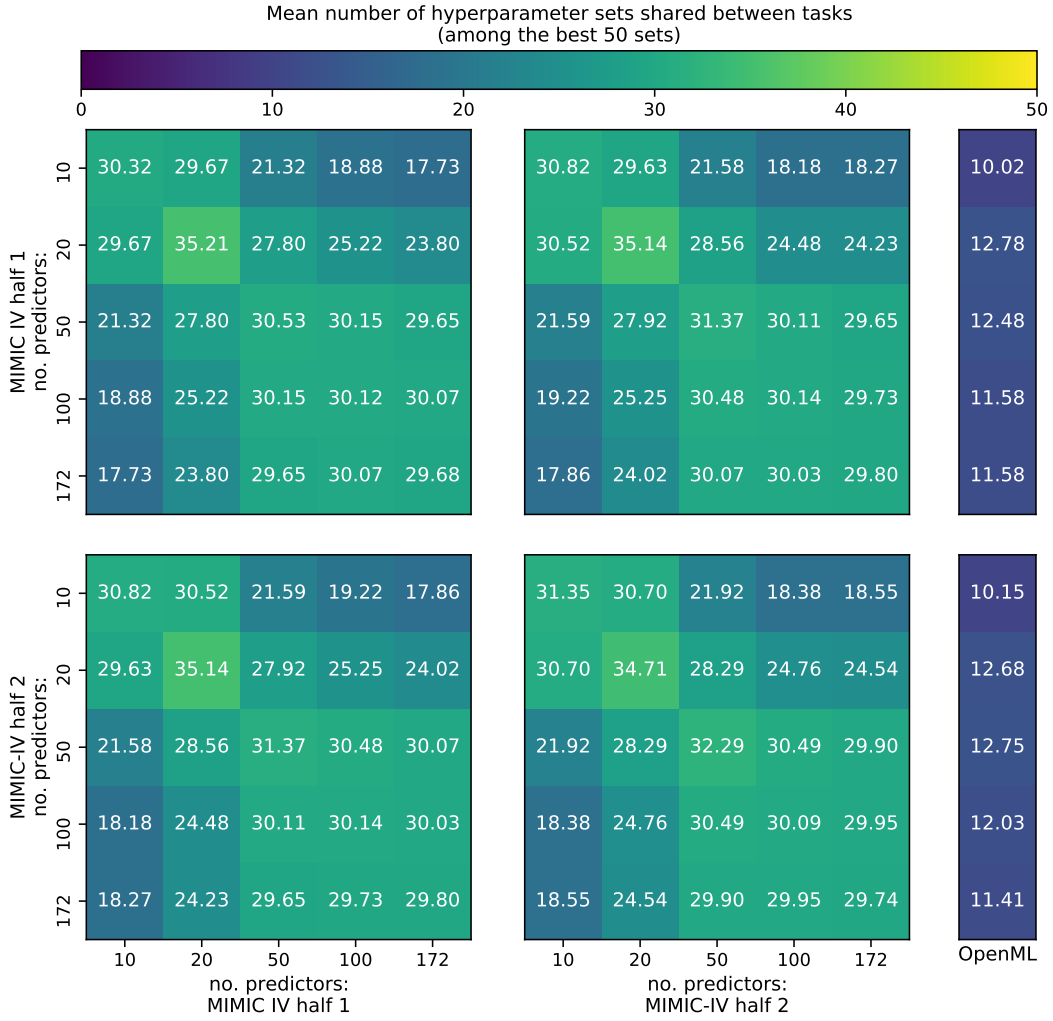

Figure 5: Summary of mean numbers of the best 50 hyperparameter sets (regarding the mean 4-CV ROC AUC measure) shared between tasks from Experiment 3. A single cell of the matrix represents the average value for tasks with a given number of columns and based on a given subset of observations. Additionally, the vectors on the right correspond to the same operation for the intersection of MIMIC-IV-based tasks with tasks derived from OpenML.

The analysis of the results of Experiment 3 required a partial aggregation of the calculated statistics because, without this, the number of possible combinations would become too large for clear representation on a graph. We decided to perform this aggregation by grouping the tasks based on their source and, for MIMIC-IV, also on the number of predictors and the considered subset of observations (Figure 5). Therefore, a single cell corresponds to the average value of a matrix created in the same way as in the previous graph.

As the number of predictors decreases, their diversity between tasks increases, which is due to the procedure of selecting them described in Experiment 3. Despite this, the average number of shared hyperparameter sets for tasks based on a similar number of predictors is consistently high. This suggests that the structural similarity of tasks (understood as the size and origin of the experiment matrix) is related to the transferability of hyperparameters. Nonetheless, even in the worst case, the average number of shared hyperparameter sets is higher between pairs of MIMIC-IV based tasks than when intersecting MIMIC-IV based tasks with tasks derived from OpenML.

5 Is hyperparameter transferability useful?

We have shown that there exists a relationship between structural similarity of tasks and transferability of optimal hyperparameters, but what relevance does this have for practical applications? As a proof-of-concept, we have performed hyperparameter tuning simulations using commonly used random search, and Bayesian optimisation [11], as well as methods using historical model tuning results for other tasks.

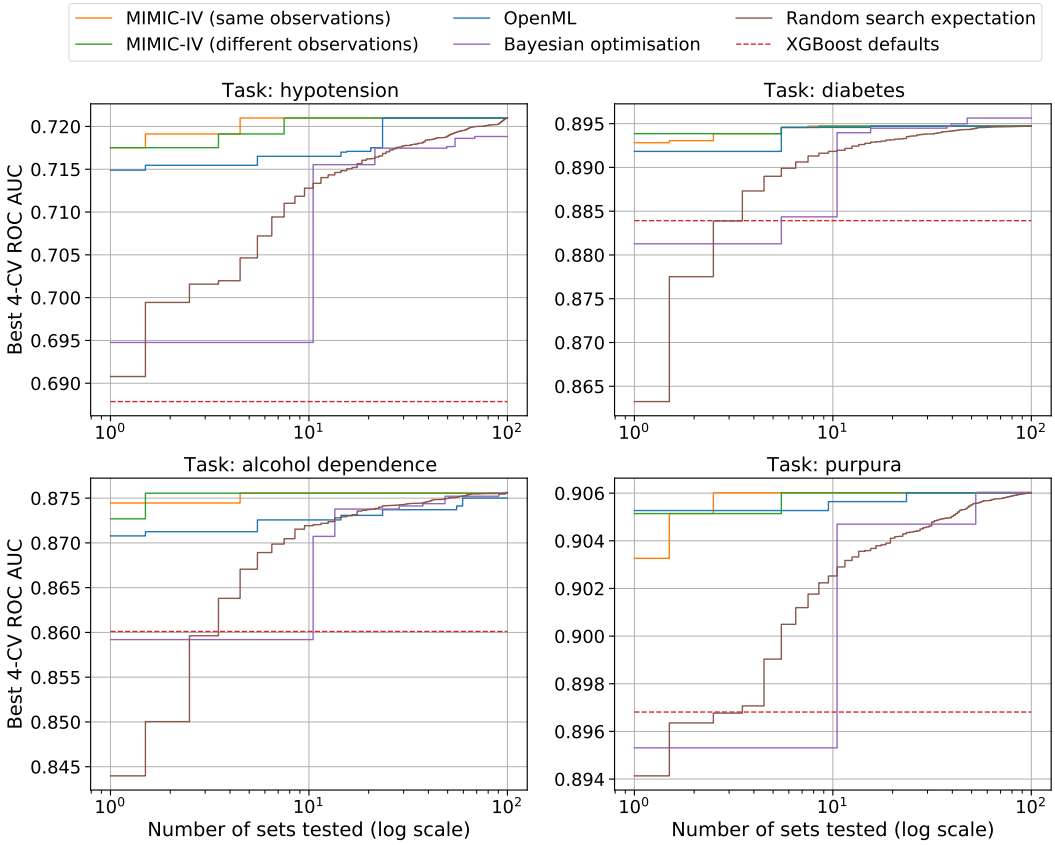

Figure 6: Hyperparameter tuning velocity of different methods and multiple benchmark tasks. Purpura is the only task for which OpenML initially significantly outperforms MIMIC-IV among the 12 tasks considered.

Figure 6 shows hyperparameter tuning velocity (best performance obtained so far) of different methods and four benchmark tasks. The curve representing XGBoost defaults does not involve tuning and only shows the results of a model based on settings predefined in the algorithm implementation used. The mean velocity of randomly searching the grid used was determined by the expected value of the beta distribution with the relevant parameters and empirical quantiles. Bayesian optimisation was performed using the implementation available in the scikit-optimize package and based on uniform distributions of hyperparameters, with bounds corresponding to the mementoML grid.

Apart from the results for the default settings, random search and Bayesian optimisation, the other three curves show the results for the proposed methods involving searching sets of hyperparameters according to a ranking built on previous model evaluations. This ranking is constructed by summing up the normalised (for each task individually) mean 4-CV ROC AUC values of models based on a given set of hyperparameters, excluding the task for which the ranking is created, and then sorting those sets in a descending order using these sums.

The results obtained confirm that tuning the model can significantly improve its performance over the default settings. It can be seen that the proposed method involving hyperparameter transfer performs better than random search and Bayesian optimisation even when the searched ranking is based on tasks unrelated to the problem under consideration. Typically, however, a ranking based on related MIMIC-IV tasks provides a slightly higher speed relative to a ranking based on OpenML, suggesting a link between the transferability strength of hyperparameters and the speed of tuning methods based on them.

6 Conclusions

The results prepared and their analysis presented in this article confirm that the structural similarity of tasks is positively related to hyperparameters' transferability between them. Notably, the conducted research uses non-synthetic data from a real-world source and is the first approach to creating a domain-based repository for meta-learning. A side effect of using the MIMIC-IV database is the research's high experimentality due to the lack of explicitly defined relationships between prediction tasks.

Our work confirms the intuition from previous studies [3, 13, 14] – hyperparameters do indeed transfer better for more similar tasks. This has broad practical implications, as exemplified by the use case scenario we gave earlier (Methodology) describing the situation of a data scientist working in a hospital.

The transferability of hyperparameters of ML models is not a well-studied area, and there is a strong need for further research on a whole range of related issues. In our opinion, these include the following topics:

- How to define and measure the similarity of predictive tasks? What description of similarity (simple or complex) best captures the capabilities of transfer learning?
- What effect do data origin and different numbers of observations considered have on transferability?
- Do the hyperparameters derived from our prepared tasks transfer to analogous tasks based on another medical domain database?
- Are the conclusions we obtained adequate for other boosting algorithms?

References

[1] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[2] T. Chen and C. Guestrin. Xgboost: A scalable tree boosting system. *CoRR*, abs/1603.02754, 2016.

[3] M. Feurer, J. T. Springenberg, and F. Hutter. Initializing bayesian hyperparameter optimization via meta-learning. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA*, pages 1128–1135. AAAI Press, 2015.

[4] A. Fisher, C. Rudin, and F. Dominici. All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously. *J. Mach. Learn. Res.*, 20:177:1–177:81, 2019.

[5] T. M. Hospedales, A. Antoniou, P. Micaelli, and A. J. Storkey. Meta-learning in neural networks: A survey. *CoRR*, abs/2004.05439, 2020.

[6] A. Johnson, L. Bulgarelli, T. Pollard, S. Horng, L. Celi, and R. Mark. MIMIC-IV (version 1.0), 2020.

[7] A. E. W. Johnson, T. J. Pollard, and R. G. Mark. Reproducibility in critical care: a mortality prediction case study. In *Proceedings of the Machine Learning for Health Care Conference, MLHC 2017, Boston, Massachusetts, USA, 18-19 August 2017*, volume 68 of *Proceedings of Machine Learning Research*, pages 361–376. PMLR, 2017.

[8] W. Kretowicz and P. Biecek. MementoML: Performance of selected machine learning algorithm configurations on OpenML100 datasets. *CoRR*, abs/2008.13162, 2020.

[9] C. Meng, L. Trinh, N. Xu, and Y. Liu. MIMIC-IF: interpretability and fairness evaluation of deep learning models on MIMIC-IV dataset. *CoRR*, abs/2102.06761, 2021.

[10] S. Purushotham, C. Meng, Z. Che, and Y. Liu. Benchmarking deep learning models on large healthcare datasets. *J. Biomed. Informatics*, 83:112–134, 2018.

[11] J. Snoek, H. Larochelle, and R. P. Adams. Practical bayesian optimization of machine learning algorithms. *CoRR*, abs/1206.2944, 2012.

[12] J. Vanschoren, J. N. van Rijn, B. Bischl, and L. Torgo. OpenML: networked science in machine learning. *SIGKDD Explor.*, 15(2):49–60, 2013.

[13] M. Wistuba, N. Schilling, and L. Schmidt-Thieme. Learning hyperparameter optimization initializations. In *2015 IEEE International Conference on Data Science and Advanced Analytics, DSAA 2015, Campus des Cordeliers, Paris, France, October 19-21, 2015*, pages 1–10. IEEE, 2015.

[14] M. Wistuba, N. Schilling, and L. Schmidt-Thieme. Sequential model-free hyperparameter tuning. In *2015 IEEE International Conference on Data Mining, ICDM 2015, Atlantic City, NJ, USA, November 14-17, 2015*, pages 1033–1038. IEEE Computer Society, 2015.

Appendix A – target correlation

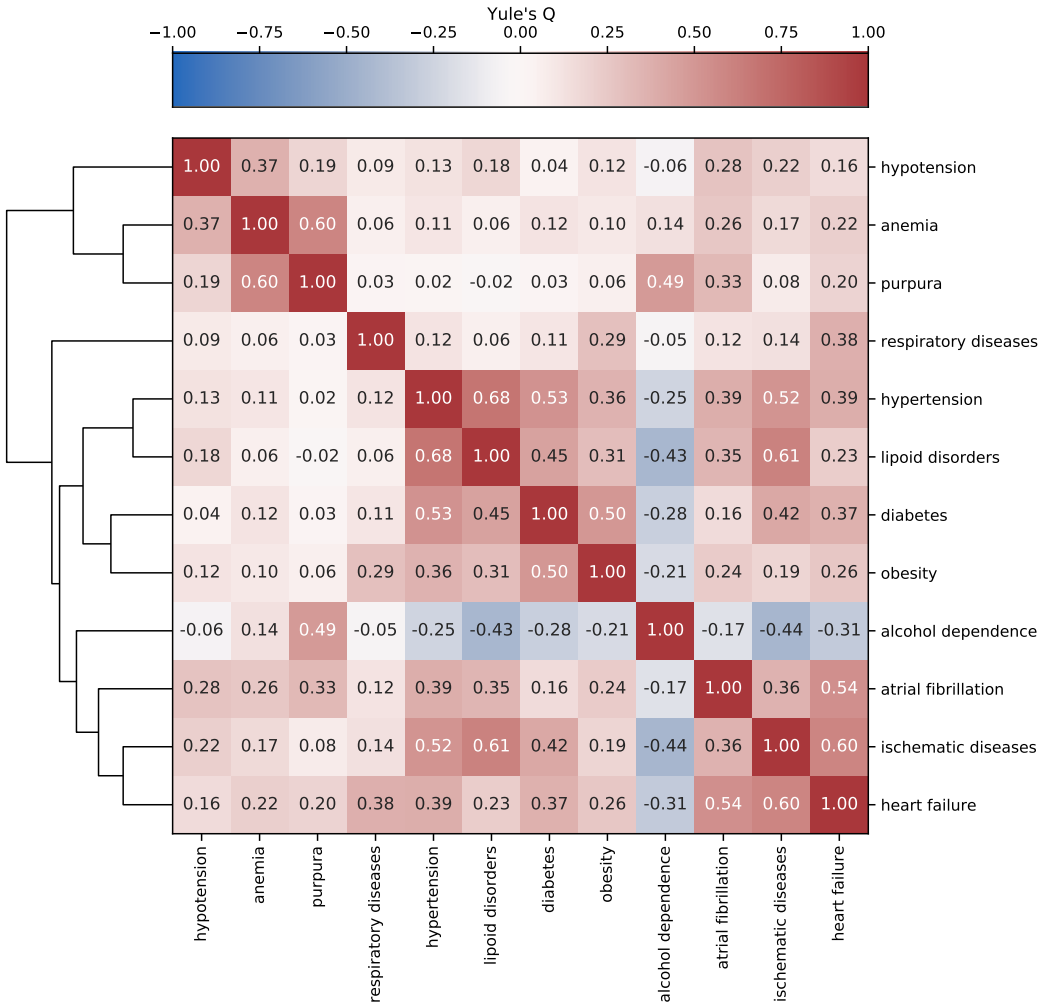

Figure 7: Yule's Q for target pairs. The dendrogram is based on Ward's hierarchical clustering with distance measure defined as $1 - |value|$.

Appendix B – selected predictors

Table 3: Selected predictors. Statistics stands for minimum, average and maximum value.

ID	Label	Category	Table	Aggregation	Missing
226253	SpO2 Desat Limit	Alarms	chartevents	statistics	0.7%
226512	Admission Weight (Kg)	General	chartevents	first value	0.1%
226730	Height (cm)	General	chartevents	first value	45.3%
220228	Hemoglobin	Labs	chartevents	statistics	2.3%
220546	WBC	Labs	chartevents	statistics	2.3%
225624	BUN	Labs	chartevents	statistics	2.1%
227073	Anion gap	Labs	chartevents	statistics	2.1%
227457	Platelet Count	Labs	chartevents	statistics	2.3%
227465	Prothrombin time	Labs	chartevents	statistics	12.5%
227466	PTT	Labs	chartevents	statistics	13.0%
220739	GCS - Eye Opening	Neurological	chartevents	statistics	0.2%
223900	GCS - Verbal Response	Neurological	chartevents	statistics	0.2%
223901	GCS - Motor Response	Neurological	chartevents	statistics	0.2%
223791	Pain Level	Pain/Sedation	chartevents	statistics	10.7%
220210	Respiratory Rate	Respiratory	chartevents	statistics	0.1%
220277	O2 saturation pulseoxymetry	Respiratory	chartevents	statistics	0.1%
223834	O2 Flow	Respiratory	chartevents	statistics	24.3%
220045	Heart Rate	Routine Vital Signs	chartevents	statistics	0.0%
220179	Non Invasive Blood Pressure systolic	Routine Vital Signs	chartevents	statistics	1.1%
220180	Non Invasive Blood Pressure diastolic	Routine Vital Signs	chartevents	statistics	1.1%
223761	Temperature Fahrenheit	Routine Vital Signs	chartevents	statistics	1.6%
224054	Braden Sensory Perception	Skin - Assessment	chartevents	statistics	0.6%
224055	Braden Moisture	Skin - Assessment	chartevents	statistics	0.6%
224056	Braden Activity	Skin - Assessment	chartevents	statistics	0.6%
224057	Braden Mobility	Skin - Assessment	chartevents	statistics	0.6%
224058	Braden Nutrition	Skin - Assessment	chartevents	statistics	0.6%
224059	Braden Friction/Shear	Skin - Assessment	chartevents	statistics	0.6%
50802	Base Excess	Blood Gas	labevents	statistics	34.6%
50804	Calculated Total CO2	Blood Gas	labevents	statistics	34.6%
50813	Lactate	Blood Gas	labevents	statistics	33.2%
50818	pCO2	Blood Gas	labevents	statistics	34.6%
50820	pH	Blood Gas	labevents	statistics	32.5%
50821	pO2	Blood Gas	labevents	statistics	34.6%
50861	Alanine Aminotransferase (ALT)	Chemistry	labevents	statistics	38.0%
50863	Alkaline Phosphatase	Chemistry	labevents	statistics	38.7%
50868	Anion Gap	Chemistry	labevents	statistics	0.5%
50878	Asparate Aminotransferase (AST)	Chemistry	labevents	statistics	37.9%
50882	Bicarbonate	Chemistry	labevents	statistics	0.5%
50885	Bilirubin, Total	Chemistry	labevents	statistics	38.7%
50893	Calcium, Total	Chemistry	labevents	statistics	3.2%
50902	Chloride	Chemistry	labevents	statistics	0.5%
50912	Creatinine	Chemistry	labevents	statistics	0.4%
50931	Glucose	Chemistry	labevents	statistics	0.5%
50960	Magnesium	Chemistry	labevents	statistics	0.9%
50970	Phosphate	Chemistry	labevents	statistics	3.1%
50971	Potassium	Chemistry	labevents	statistics	0.5%
50983	Sodium	Chemistry	labevents	statistics	0.5%
51006	Urea Nitrogen	Chemistry	labevents	statistics	0.4%
51221	Hematocrit	Hematology	labevents	statistics	0.5%
51222	Hemoglobin	Hematology	labevents	statistics	0.5%
51248	MCH	Hematology	labevents	statistics	0.5%
51249	MCHC	Hematology	labevents	statistics	0.5%
51250	MCV	Hematology	labevents	statistics	0.5%
51265	Platelet Count	Hematology	labevents	statistics	0.5%
51277	RDW	Hematology	labevents	statistics	0.5%
51279	Red Blood Cells	Hematology	labevents	statistics	0.5%
51301	White Blood Cells	Hematology	labevents	statistics	0.5%
51491	pH	Hematology	labevents	statistics	41.5%