Feature selection algorithm recommendation for gene expression data through gradient boosting and neural network metamodels

Robert Aduviri
Pontifical Catholic University of Peru
robert.aduviri@pucp.edu.pe

Daniel Matos
Pontifical Catholic University of Peru
daniel.matos@pucp.pe

Edwin Villanueva
Pontifical Catholic University of Peru
ervillanueva@pucp.edu.pe

***Abstract*—Feature selection is an important step in gene expression data analysis. However, many feature selection methods exist and a costly experimentation is usually needed to determine the most suitable one for a given problem. This paper presents the application of gradient boosting and neural network techniques for the construction of metamodels that can recommend rankings of {feature selection - classification} algorithm pairs for new gene expression classification problems. Results in a corpus of 60 public data sets show the superiority of these techniques in producing more useful rankings in relation to classical metamodels.**

***Keywords*—Metalearning, gene expression data, feature selection, cancer classification, algorithm recommendation.**

I. Introduction

The current gene expression profiling technologies have generated great hopes for the construction of early diagnosis and prognosis systems for cancer and other diseases. However, building such systems with clinically acceptable accuracies is challenging. Among the major difficulties is the high dimensionality of the feature space (genes) relative to the number of samples, which poses statistical issues in model estimation [1]. The usual approach to alleviate this problem is by applying feature selection techniques. Yet many feature selection methods exist and none is clearly superior in the domain of gene expression data [1]. So the common practice is to experiment with a set of selection methods in combination with classification models to determine the most suitable for a given problem (gene expression dataset), implying high experimentation times and computational resources. Metalearning (MtL) has been proposed as a way to circumvent this costly practice [2]. It aims to construct predictive models (metamodels) that relate characteristics of datasets (metafeatures) to performance of algorithms so they can be used to recommend algorithms for new unseen problems. Nonetheless, MtL has been little exploited in the domain of gene expression data. Representative works are [3] for clustering algorithm recommendation and [4], [5] for classification algorithm recommendation. Classical machine learning models have been used as metamodels, including KNN, SVM and Ranking Trees, which have shown some potential of MtL in this domain. However, more sophisticated and robust models exist, such as gradient boosting machines (GBM) [6] and neural networks, which could make MtL approaches more attractive in this field. Thus, in this paper we adapt and evaluate two state-of-the-art ML models for ranking recommendation of {feature selection - classification} algorithm pairs for gene expression classification problems: a GBM with the LambdaRank ranking cost function [7] and a neural network model.

This work has been supported by INNOVATE PERU (Grant 334-INNOVATEPERU-BRI-2016) .

II. Methods

This work follows the general MtL scheme in [8] (Fig. 1). The input consists in a repository of datasets from different problems. The data characterization module extracts metafeatures that describe each dataset. The evaluation module assess the performance scores of each considered algorithm in each dataset. These scores are then converted to rankings that, together with the metafeatures, form the metadata for the metamodel induction. A ML algorithm can then be used to induce the metamodel with the metafeatures as input variables and the ranking as target variables. This metamodel can be used in testing time to predict a ranking of algorithms for a new problem, based on the metafeatures of the input dataset.

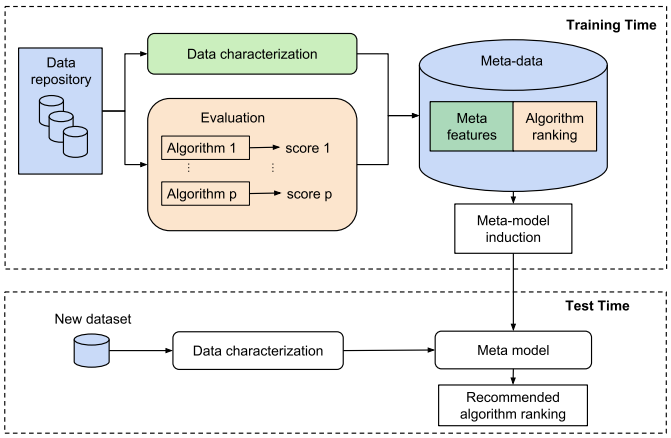

Fig. 1. Metalearning for algorithm recommendation (adapted from [8])

In this work we focus on the metamodel induction module aiming to feature selection algorithm recommendation. The first evaluated method is the LightGBM (LGBM) algorithm

[6], an ensemble of gradient boosting decision trees with the LambdaRank pairwise loss function. It has shown successful results in real world ranking problems [7], because it allows to optimize the Normalized Discounted Cumulative Gain (NDCG) metric by adjusting the learning rate based on the NDCG changes obtained by swapping two data points. In this case, the optimization metric used for early stopping and parameter tuning was NDCG@n, where n is the number of all the feature selection methods available to recommend. In other words, the NDCG was calculated over the whole ranking list. LightGBM works over discrete features, continuous features are transformed into discrete ones by histogram binning. In order to find the best configuration of parameters of the estimators and the training procedure, hyperparameter tuning is performed with Bayesian optimization [9].

As an alternative method to build ranking metamodels we propose a neural network architecture inspired on a matrix factorization approach. The architecture is illustrated in Fig. 2, where the feature selection method is transformed to a dense representation via a embedding layer, while the metafeatures, already in a continuous representation are transformed to the same latent space via a dense layer with a nonlinear activation. Once in the latent space, both representations are combined with a dot product to ensure that they share the same latent space. The output of the network consists in a continuous value that represents the relative rank of the algorithm, normalized between 0 and 1 via a sigmoid function, which is optimized through a MSE loss function.

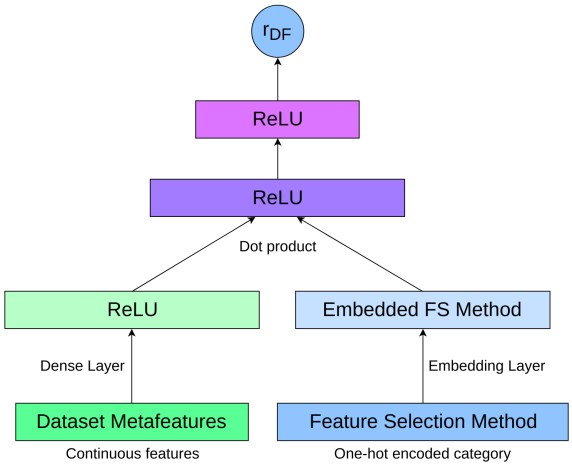

Fig. 2. Neural network architecture. Both the continuous metafeatures and the feature selection method indices are transformed to the same latent space via dense and embedding layers and then merged through a dot product

As a baseline we also evaluate the classic K-Nearest Neighbors (KNN) algorithm as a ranking recommendation method [4]. This method constructs the ranking recommendation by averaging the rankings of the k nearest datasets to the test dataset according to a distance measure. The average ranking of all the training rankings is also evaluated as a baseline.

III. Experiments and Results

For the experiments of this work we used a collection of 60 public gene expression datasets derived from different cancer-related studies. These datasets can be found in our repository[1]. Each dataset was evaluated with every combination of 4 feature selection algorithms (ReliefF, Fisher-score, Chi2 and Random Forest) and 3 classification methods (Support Vector Machines, Naive Bayes and Logistic Regression). The evaluation was performed in a 5-fold cross-validation strategy, repeated 30 times with different foldings. The average Gmean (geometric mean of class-specific accuracies) was used as a score of each combination, which was used to construct the target ranking. As metafeatures we used 12 common statistics and based on information theory measures [8] which we expanded to 39 using the framework for systematic development of metafeatures proposed by [10].

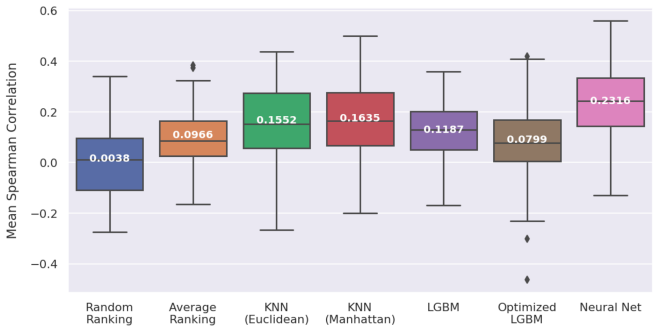

Fig. 3. Spearman correlation results of different metamodel induction methods. The white labels display the mean value

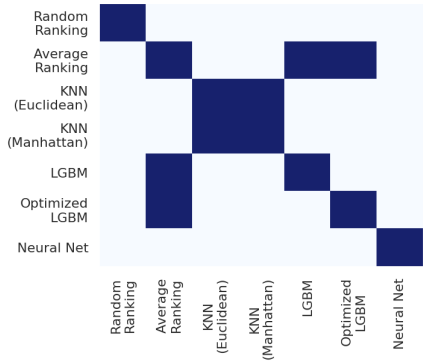

Fig. 4. Paired statistical test results on the mean Spearman scores of the different metamodels. Blue boxes indicate pairs of metamodels with not statistically significant difference (significance level of 0.05).

To evaluate the quality of the recommended rankings we use two measures: the Spearman correlation index [8] and the performance loss curve (PLC) metric [11]. The Spearman index assesses the overall proximity of the estimated ranking w.r.t. the ideal ranking. The PLC metric evaluates how useful the ranking is (in terms of accuracy) if one evaluate the algorithms in the ranking order. To compute PLC, each algorithm of the ranking is sequentially tested and the difference in accuracy between the best algorithm so far and the truly best algorithm is stored. Then, a loss curve is generated with that differences and the area under that curve is the PLC metric. The score of

[1]https://github.com/Howl24/fs-ranking-prediction

each metamodel induction method was obtained by averaging the PLC results of 10-fold, 10-times cross-validation over the metadata.

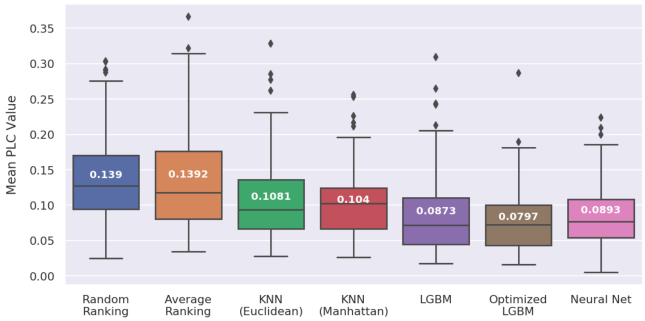

Fig. 5. PLC results of different metamodel induction methods. The white labels display the mean value

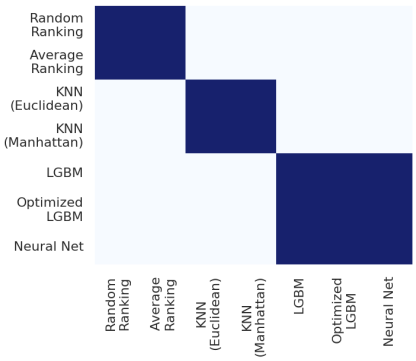

Fig. 6. Paired statistical test results on the mean PLC scores of the different metamodels. Blue boxes indicate pairs of metamodels with not statistically significant difference (significance level of 0.05)

Fig. 3 shows the average Spearman scores of the metamodels induced with each evaluated method. We can observe that Neural Networks metamodels present the best mean scores, followed by KNN and LGBM metamodels. The worst performing method was random ranking, as expected. To assess the statistical significance of these results we applied a paired T-test and a Wilcoxon test to test the mean differences between all pairs of metamodel Spearman correlation and PLC scores respectively (with a significance level of 0.05). Fig 4. shows the results of these tests. It can be noted that neural networks metamodels are statistically different from the other models, which confirms their great learning capacity to suggest overall rankings.

Fig. 5 shows the average PLC scores of the metamodels induced with each evaluated method, and Fig 6. shows the corresponding results of the statistical significance tests among all pairs of metamodel scores, as explained above. These results indicate that any MtL approach is statistically better (in terms of PLC values) than choosing random rankings or the average ranking of the training datasets. Among the learned metamodels, we can observe that KNN produce rankings with statistically worse PLC scores than LGBM or Neural Network metamodels, as we hypothesized, since these last are more elaborated. LGBM and Neural Network metamodels tend to offer similar scores and variances and do not show significant statistical differences. However, the LGBM optimized version improves slightly the average PLC results.

It is important to say that PLC metric is a more useful measure for the intended task than Spearman index, since it give us an idea of how much we can gain or lose in accuracy if we follow the recommended ranking to build the classifiers. The Spearman index evaluates the overall proximity of the inferred ranking to the ideal one, giving the same weight to errors in the higher or lower part of the ranking and without worrying about predictive accuracy of the base-level models.

IV. Conclusion

This paper adapted and evaluated two state-of-the-art methods to predict rankings of combinations of feature selection - classification algorithms for gene expression classification problems. Results on a collection of 60 public gene expression datasets showed a significant gain in prediction accuracy and stability in relation to the standard KNN method, as measured by the PLC metric. Further improvements were obtained when an optimization effort was made in these models. This proves that, by taking advantage of current developments in the machine learning field, we can improve our ability to deal with these challenging data and to facilitate the construction of early diagnosis and prognosis systems for cancer without incurring in high computational burden.

References

[1] A. Bhola and S. Singh, "Gene selection using high dimensional gene expression data: An appraisal," *Curr. Bioi.*, vol. 13, pp. 225–233, 2018.

[2] P. Brazdil and C. Giraud-Carrier, "Metalearning and algorithm selection: progress, state of the art and introduction to the 2018 special issue," *Machine Learning*, vol. 107, no. 1, pp. 1–14, 2018.

[3] M. Vukicevic, S. Radovanovic, B. Delibasic, and M. Suknovic, "Extending meta-learning framework for clustering gene expression data with component-based algorithm design and internal evaluation measures," *IJDMB*, vol. 14, no. 2, pp. 101–119, 2016.

[4] B. F. de Souza, C. Soares, and A. C. Carvalho, "Meta-learning approach to gene expression data classification," *Int. J. of Intel. Comp. and Cyber.*, vol. 2, no. 2, pp. 285–303, 2009.

[5] B. F. de Souza, A. C. de Carvalho, and C. Soares, "Empirical evaluation of ranking prediction methods for gene expression data classification," in *IBERAMIA 2010*. Springer Berlin Heidelberg, 2010, pp. 194–203.

[6] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "Lightgbm: A highly efficient gradient boosting decision tree," in *Adv. in Neural Inf. Proces. Systems 30*, 2017, pp. 3146–3154.

[7] F. Yuan, G. Guo, J. M. Jose, L. Chen, H. Yu, and W. Zhang, "Lambdafm: Learning optimal ranking with factorization machines using lambda surrogates," in *25th ACM CIKM '16*, 2016, pp. 227–236.

[8] P. Brazdil, C. Giraud-Carrier, C. Soares, and R. Vilalta, *Metalearning: Applications to Data Mining*, 1st ed. Springer, 2008.

[9] J. Snoek, H. Larochelle, and R. P. Adams, "Practical bayesian optimization of machine learning algorithms." in *NIPS*, P. L. Bartlett, F. C. N. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds., 2012, pp. 2960–2968. [Online]. Available: http://dblp.uni-trier.de/db/conf/nips/nips2012.htmlSnoekLA12

[10] F. Pinto, C. Soares, e. J. Mendes-Moreira, João", L. Khan, T. Washio, G. Dobbie, J. Z. Huang, and R. Wang, ""towards automatic generation of metafeatures," in *PAKDD 2016*, 2016, pp. 215–226.

[11] S. M. Abdulrahman, P. Brazdil, J. N. Van Rijn, and J. Vanschoren, "Algorithm selection via meta-learning and sample-based active testing," in *MetaSel'15*, 2015, pp. 55–66.