2022 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB) | 978-1-6654-8462-6/22/$31.00 ©2022 IEEE | DOI: 10.1109/CIBCB55180.2022.9863032

Compound Virtual Screening by Learning-to-Rank with Gradient Boosting Decision Tree and Enrichment-based Cumulative Gain

Kairi Furui
Department of Computer Science
School of Computing
Tokyo Institute of Technology
Kanagawa, Japan
furui@li.c.titech.ac.jp

Masahito Ohue
Department of Computer Science
School of Computing
Tokyo Institute of Technology
Kanagawa, Japan
ohue@c.titech.ac.jp

***Abstract*—Learning-to-rank, a machine learning technique widely used in information retrieval, has recently been applied to the problem of ligand-based virtual screening to accelerate the early stages of new drug development. Ranking prediction models learn based on ordinal relationships, making them suitable for integrating assay data from various environments. Existing studies of rank prediction in compound screening have generally used a learning-to-rank method called RankSVM. However, they have not been compared with or validated against the gradient boosting decision tree (GBDT)-based learning-to-rank methods that have gained popularity recently. Furthermore, although the ranking metric called Normalized Discounted Cumulative Gain (NDCG) is widely used in information retrieval, it only determines whether the predictions are better than those of other models. In other words, NDCG cannot recognize when a prediction model produces worse than random results. Nevertheless, NDCG is still used in the performance evaluation of compound screening using learning-to-rank. This study used the GBDT model with ranking loss functions, called lambdarank and lambdaloss, for ligand-based virtual screening; results were compared with existing RankSVM methods and GBDT models using regression. We also proposed a new ranking metric, Normalized Enrichment Discounted Cumulative Gain (NEDCG), aiming to evaluate the goodness of ranking predictions properly. In addition, the results showed that the GBDT model with learning-to-rank outperformed existing regression methods using GBDT and RankSVM on diverse datasets. Finally, NEDCG showed that the predictions by regression were comparable to random predictions in multi-assay, multi-family datasets, demonstrating its usefulness for a more direct assessment of compound screening performance.**

***Index Terms*—Drug discovery, Cheminformatics, Learning-to-rank, Machine learning, Virtual screening**

I. Introduction

The development cost and time required to obtain approval for a new drug increase every year, with some estimating a cost of $2.6 billion per drug [1], and others reporting a development time of more than 10 years from identifying lead compounds to clinical trials [2]. Virtual screening (VS) is a process of computationally searching an extensive compound library for an active compound against a target protein in the early stage of new drug development. Virtual screening technology helps in the discovery process of hit compounds [3]. For virtual screening in the drug repositioning context, the drug Edaravone (Radicava), FDA-approved for cardiovascular indications, was identified as a neurotrophic [4]. Virtual screening with the FDA-approved drug dataset was also implemented for the Coronavirus Disease-2019 (COVID-19) virus, which has caused a pandemic since December 2019 [5]. Improving the predictive accuracy of virtual screening and extending its applicability to a wide range of activity data are essential to reducing the cost of developing new drugs.

Ligand-based virtual screening (LBVS) is a method that uses activity information already obtained from assays and mainly employs machine learning methods such as regression [6], [7] and classification [8]–[10] prediction. Recently, a learning-to-rank method based on the ordinal relationship of activity values was proposed for virtual screening [11]–[18]. This approach has two advantages, of which the first is that learning-to-rank is more accurate than regression for ordinal predictions [15]. In drug discovery, potentially active compounds selected through virtual screening are biochemically assayed to determine whether they are active. Therefore, the goal of virtual screening is not to predict the exact activity value but to list compounds that are even slightly more active at the top of the prediction. For this reason, learning-to-rank, predicting based on order, is appropriate for virtual screening. The second advantage is that learning-by-rank is dependent on ordering relationships in comparable groups, such as assay, making it easy to integrate experimental information from different situations. Affinity indices based on biochemical assays, such as half the maximum inhibitory concentration (IC_{50}), vary widely from one assay system to another. This aspect makes it challenging to integrate assay data from different environments using regression methods. In learning-to-rank, the distribution of measurements in each assay need not be identical because the model learns based on the ordinal relationship among compounds within each assay [15].

However, there are several problems with existing studies

of virtual screening using learning-to-rank. First, existing studies [13], [15]–[18] mainly focus on RankSVM, a machine learning method used for ranking prediction, and do not evaluate the effectiveness of new methods. Recently, a learning-to-rank method called LambdaMART has been broadly used in the information retrieval field [19]. LambdaMART is a machine learning method that learns a gradient boosting decision tree (GBDT) with a ranking loss function. This method has attracted attention in machine learning competitions. No comparisons have been reported regarding whether LambdaMART is superior to RankSVM or whether learning-to-rank is superior to regression in order prediction, even in the GBDT model, for virtual screening. Furthermore, previous studies [15]–[17] have evaluated the performance of ranking prediction using a metric called Normalized Discounted Cumulative Gain (NDCG) [20]–[22], but the suitability of NDCG for LBVS has not been adequately discussed. NDCG is a metric designed for use in learning-to-rank performance evaluation in information retrieval, which differs from the situation in virtual screening. Specifically, NDCG reports a maximum value of 1 to predict that a sequence is perfectly correct. However, the value reported by NDCG does not express the extent to which a prediction represents an improvement. In LBVS, enrichment, i.e., the improvement degree relative to the random prediction is essential. However, existing studies use NDCG as a metric for evaluating virtual screening against ranking prediction without considering these differences.

In this study, we evaluate the VS performance of a new GBDT model with learning-to-rank and compare the performance with existing models. Moreover, we develop a metric that can evaluate the performance of ranking prediction appropriately. The three main contributions of this study are:

- We modified the conventional NDCG and proposed NEDCG (normalized enrichment DCG), a metric expressing the improvement ratio over random guess in ranking prediction.
- We applied for the first time the GBDT model with lambdarank loss function to the virtual screening problem.
- We validated NEDCG based on a dataset constructed for multiple situations where assay data was available. The results showed that the prediction accuracy of the proposed method using GBDT outperformed that of the conventional method based on learning-to-rank. In addition, the validation indicated that in some cases, the learning-to-rank was more effective. In others cases, the regression models were more accurate in prediction, depending on the dataset.

II. Materials and Methods

A. GBDT and LambdaMART

GBDT is a machine learning algorithm that minimizes the cost function by iteratively ensembling weak prediction models using decision trees. Moreover, GBDT is a widely used algorithm, and several effective implementations exist, including XGBoost [23] and LightGBM [24]. In this study, we used LightGBM as an implementation of GBDT.

LambdaMART [19] is a method for learning GBDT with a ranking loss function called lambdarank [25], designed to directly optimize the value of NDCG [20]–[22]. The NDCG for the top K cases from a group of N cases is as follows:

$$\begin{aligned} \mathrm{NDCG@}K &= \sum_{i=1}^{K} \frac{G_i}{D_i}, \\ G_i &= \frac{gain_i}{\mathrm{maxDCG}}, \\ gain_i &= 2^{y_i} - 1, \\ D_i &= \log_2(i+1), \end{aligned}$$

where i is the predicted rank, y_i is the label at i, and maxDCG is a normalization constant, which is the maximum discounted cumulative gain (DCG) when the ranking prediction is correct. If the top of the ranking is correctly ordered, NDCG approaches its maximum value of 1, while at its worst, it has a minimum value of 0. The loss of lambdarank for each group is as follows:

$$\begin{aligned} l_1(\boldsymbol{y}, \boldsymbol{s}) &= \sum_{y_i > y_j} \rho_{ij} \left| G_i - G_j \right| \log(1 + e^{-\sigma(s_i - s_j)}), \\ \rho_{ij} &= \left| \frac{1}{D_i} - \frac{1}{D_j} \right|, \end{aligned}$$

where s_i and s_j are the predicted ranking scores. Lambdarank learns order relationships by penalizing based on $\Delta\mathrm{NDCG}_{ij} = \rho_{ij}|G_i - G_j|$, i.e., the difference in NDCG when rank i is swapped with rank j.

Moreover, we experimented with a loss function based on the lambdaloss framework, called NDCGloss2 [26], which minimizes a cost function called $\mathrm{NDCG}_{\mathrm{cost}}$. The lambdaloss function for each group and $\mathrm{NDCG}_{\mathrm{cost}}$ is as follows:

$$\begin{aligned} l_2(\boldsymbol{y}, \boldsymbol{s}) &= \sum_{y_i > y_j} \delta_{ij} \left| G_i - G_j \right| \log(1 + e^{-\sigma(s_i - s_j)}), \\ \delta_{ij} &= \left| \frac{1}{D_{|i-j|}} - \frac{1}{D_{|i-j|+1}} \right|, \\ \mathrm{NDCG}_{\mathrm{cost}} &= \sum_{i=1}^{N} G_i - \sum_{i=1}^{N} \frac{G_i}{D_i}. \end{aligned}$$

B. Normalized enrichment DCG

NDCG is one of the primary metrics used to measure the performance of learning-to-rank models. In particular, NDCG is appropriate for comparative analysis of different models and determining which hyperparameters are better. However, this method lacks information on the degree of improvement of the ranking predictions from the pre-training situation. For example, an area under the receiver operating characteristic curve (AUROC) of 0.5, used in the classification task, implies that the model makes random predictions. Therefore, we propose a new normalized enrichment DCG (NEDCG), inspired by AUROC, removing the effect of random prediction (pre-

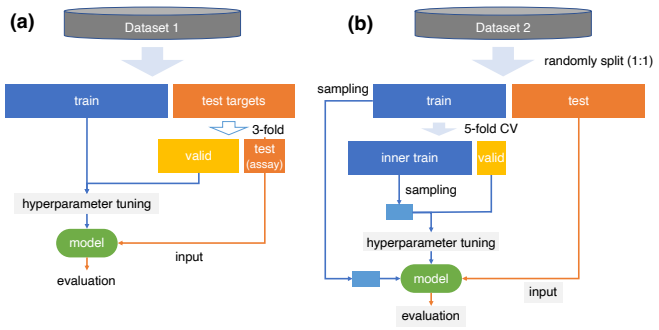


Fig. 1: Experimental procedure for (a) Dataset 1 (complicated task) and (b) Dataset 2 (simple task).

training state) from NDCG. NEDCG for the top K cases are defined as follows:

$$\mathrm{NEDCG@}K = \frac{\mathrm{DCG@}K - \mathrm{randomDCG@}K}{\mathrm{maxDCG@}K - \mathrm{randomDCG@}K}, \quad (1)$$

$$\mathrm{DCG@}K = \sum_{i=1}^{K} \frac{gain_i}{\log_2(i+1)}, \quad (2)$$

where $\mathrm{randomDCG@}K$ is the discounted cumulative gain $\mathrm{DCG@}K$ for the top K cases when predicting randomly. When calculating $\mathrm{randomDCG@}K$, the average $gain_{mean}$ of the gains for a group of N cases is used as follows:

$$\mathrm{randomDCG@}K = \sum_{i=1}^{K} \frac{gain_{mean}}{\log_2(i+1)},$$

$$gain_{mean} = \frac{1}{N} \sum_{j=1}^{N} gain_j.$$

III. Experiments

A. Dataset

In this section, we describe two datasets for different experimental situations. Dataset 1 (complicated task) includes experimental data from various assays for different proteins of the same family are used as training data. Dataset 2 (simple task) is a typical LBVS case where data from a single assay is randomly split into training and test data.

In Dataset 1, assay data for the phosphodiesterase (PDE) family were collected for IC_{50} from the ChEMBL database [27] with reference to existing studies [15], [16]. Data with the same assay ID were treated as a group, and assays with a group size of 5 or more were selected. The objective variable was IC_{50} converted to $\mathrm{pIC}_{50} = -\log_{10}(\mathrm{IC}_{50})$. For inactive data (cases with no value data were described as inactive), $\mathrm{pIC}_{50} = 0$ based on [17]. Table I shows the details of Dataset 1. In addition, Table II lists the ChEMBL Assay IDs of the assays used as test data and the number of compounds in each.

In Dataset 2, the Luciferase dataset (AID: 1006) for the inhibition rate measurement experiment was used following a previous study [17]. The Luciferase dataset was obtained from PubChem BioAssay [28], a compound activity information database. The activity value for Dataset 2 was %Inhibition. However, because the gain used in the NDCG calculation was a power of 2 of the activity value, the inhibition rate value was divided by 10 and converted so that the maximum value is 10. Note that the measured inhibition rate ranged from $-\infty$ to 100%, but the negative values were converted to set the inhibition rate to 0. Dataset 2 contains 2,976 active compounds (inhibition rate ≥ 50) and 192,588 inactive compounds (inhibition rate < 50).

TABLE I: Details of Dataset 1 (PDE family dataset). Target proteins in bold were excluded from the training step.

protein name	#assays	#compounds
PDE 1A	4	76
PDE 1B	9	101
PDE 1C	7	106
PDE 2A	41	999
PDE 3A	22	324
PDE 3B	8	121
PDE 4A	36	685
PDE 4B	69	1,457
PDE 4C	9	110
PDE 4D	57	985
PDE 5A	87	2,889
PDE 6A	1	47
PDE 6C	3	33
PDE 6D	1	24
PDE 7A	22	659
PDE 7B	2	15
PDE 8A	4	43
PDE 8B	2	168
PDE 9A	23	568
PDE 10A	79	3,848
PDE 11A	12	181

TABLE II: Dataset 1: Assays used as tests and their sizes.

	ChEMBL Assay ID	#compounds
PDE 2A	CHEMBL3706318	146
PDE 7A	CHEMBL3706063	170
PDE 8B	CHEMBL2073616	111

B. Procedure

We compare the following 4 prediction models:

1) lambdaloss (rank): GBDT ranking prediction model using the lambdaloss function
2) lambdarank (rank): GBDT ranking prediction model using the lambdarank loss function
3) RankSVM (rank): RBF Kernel RankSVM model based on PKRank [16] for Dataset 1. Linear kernel RankSVM model based on SPDRank for Dataset 2.
4) GBDT regression (regression): GBDT regression model with L_2 loss

The experiments with regression were intended to compare the use of a ranking loss function with a regression loss function. The experiments with RankSVM were intended to compare the other learning-to-rank methods with the proposed method.

We adopted RankSVM as a representative of the existing learning-to-rank models because it was the best learning-to-rank model in a previous study [15]. In Dataset 2, we used SP-

DRank, unlike Dataset 1, because training with the SPDRank model using stochastic gradient descent was successful.

C. Features

The 1-D and 2-D descriptors (1,613 dimensions) from mordred [29] (Version: 1.2.0) were used as descriptors for the compounds. However, we removed descriptors taking null values for more than half of the compounds in the training data, resulting in 1,452 dimensions in Dataset 1 and 1,447 dimensions in Dataset 2. In addition, the normalized Smith-Waterman scores with targets in the training data were used as protein features.

However, for RankSVM in Dataset 2, we used ECFP4 (2,048 bit) [30] features computed with RDKit [31] because SPDRank can learn only sparse features.

D. Training and hyperparameters

The number of rounds in the GBDT model was 100 when tuning the hyperparameters, and the number of rounds for validation was decided by `early_stopping = 1000` based on NDCG@10. The test data predictions were made using a model with 1.1 times the number of rounds determined for the validation data. For tuning the GBDT hyperparameters, the number of leaves in the decision tree `num_leaves` were (15, 31, 63, 127, 255, 511, 1023, 2047, 4095), and the minimum number of data assigned to a leaf `min_data_in_leaf` were (10, 25, 50, 100, 200). The other hyperparameters were set as follows: `lambda_l1` = 0, `lambda_l2` = 0, `feature_fraction` = 0.7, `bagging_fraction` = 1.0, and `bagging_freq` = 0. For the learning rate, we used `learning_rate` = 0.1 for hyperparameter tuning and `learning_rate` = 0.05 for all other learning.

For Dataset 1, when using the lambdarank and lambdaloss loss functions in the GBDT model, the `lambdarank_truncation__level` was fixed at 30, and the `label_gain` step width δ was fixed at 1.0. For Dataset 2, when using the lambdarank and lambdaloss loss functions in the GBDT model, the `lambdarank_truncation__level` was fixed at 200, and the `label_gain` step width δ was fixed at 0.01.

In the RankSVM experiments, the PKRank-based method used the implementation presented by Kuo *et al*. [32], and the SPDRank-based method used the implementation presented by Ohue *et al*. [17]. The PKRank method searches for the cost parameter C $(10^{-9}, 10^{-8}, ..., 10^{0})$ and the parameter γ $(10^{-6}, 10^{-5}, 10^{-4}, 10^{-3})$ using an RBF kernel.

Figure 1(a) shows the training and evaluation procedure for Dataset 1. The test data are those listed in Table II. The validation data used for hyperparameter tuning does not contain information related to the proteins in the test. For example, when predicting the CHEMBL3706318 assay of PDE 2A as test data, 22 assays of PDE 7A and 2 assays of PDE 8B were used as validation data, and the assays of PDE 2A were not used. The metric used for tuning parameters was NDCG@10.

Figure 1(b) shows the procedure for Dataset 2. For Dataset 2, data were randomly split into training and test sets in a 1 : 1 ratio. The training data size differs substantially from that of Dataset 1. Thus, the training data were sampled randomly so that the number of instances was 10000. Moreover, the parameter tuning was performed by 5-fold cross-validation, which was sampled randomly so that the number of training instances for each fold was 10000. The metric used for tuning parameters was NDCG%10, representing the performance of the top 10% of the prediction.

IV. Results

Figure 2 shows the experimental results for the 2 datasets using 2 metrics, NDCG@K and NEDCG@K. A higher score for a few samples indicates that the model can rank the active compounds at the top of the prediction.

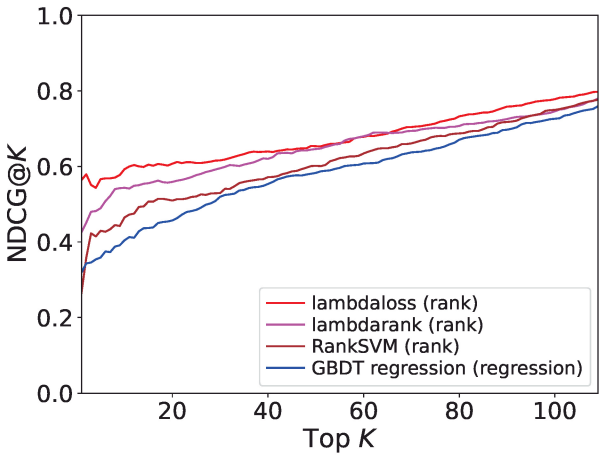
Figure 2(a) shows that the NDCG@K for the top dozen cases are lambdaloss, lambdarank, RankSVM, and regression, in that order. This result indicates that learning-to-rank with GBDT provides more appropriate insights than existing RankSVM or GBDT with regression in a compound screening of novel targets using assays with various environments, such as Dataset 1.

In Figure 2(b), for the 97,782 test samples, the regression is best for NDCG@1, while RankSVM and lambdaloss are better for NDCG@10. After that, no significant difference is present in the overall performance between lambdarank and regression. These results indicate that learning-to-rank is less effective for single assays such as Dataset 2. Moreover, this method does not consistently outperform regression methods. Note that RankSVM exhibited the lowest prediction accuracy; however, this was not an exact comparison, as RankSVM with fingerprinted features and linear kernels was used for Dataset 2.

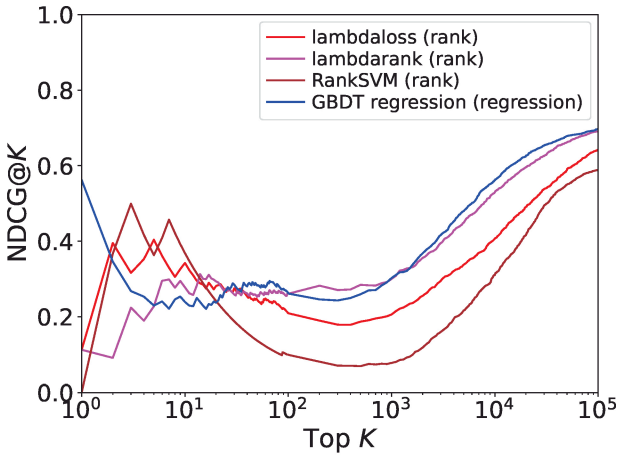
In Figure 2(c), lambdaloss shows the best NEDCG value, followed by lambdarank, RankSVM, and regression, in that order. Here, the NEDCG values for the regression method are generally equal to or less than 0.0, making it less accurate than random guessing. Therefore, a learning-to-rank method rather than regression learning is appropriate for LBVS that uses data on different assay systems and multiple proteins. As in Dataset 1, it is essential to consider how much the prediction of a new target has improved from a random prediction, and NEDCG is available for this assessment.

Figure 2(d) is almost the same as Figure 2(b). This is because the prediction is improved at least a dozen times compared to the random prediction in each model.

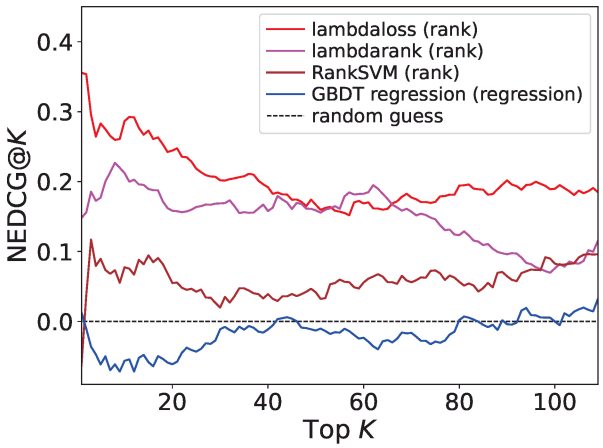
Table III summarizes the NDCG and NEDCG scores of the top 10 and top 10 % for Dataset 1 and Dataset 2 for each model, respectively. The top 10 % comprises 11 samples for Dataset 1 and 9,778 samples for Dataset 2. The scores for the top 10 samples indicate how well they predict compounds with good activity at the limited top of the prediction. The scores for the top 10 % indicate the performance of observing the top cases in the dataset.



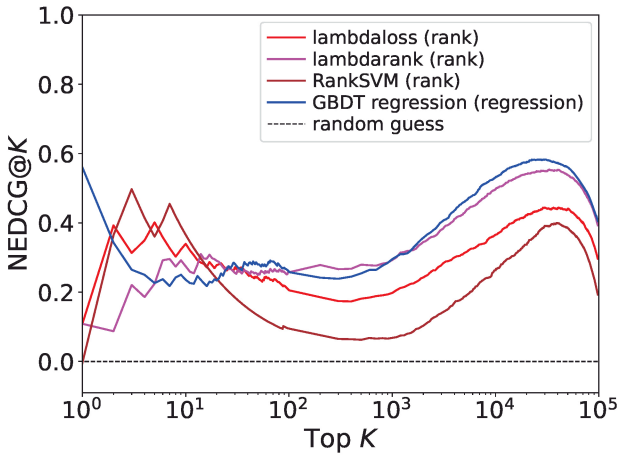
(a) Dataset 1 (complicated task), NDCG



(b) Dataset 2 (simple task), NDCG

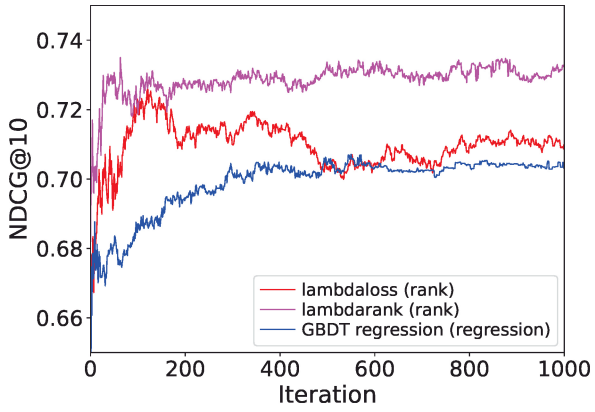


(c) Dataset 1 (complicated task), NEDCG

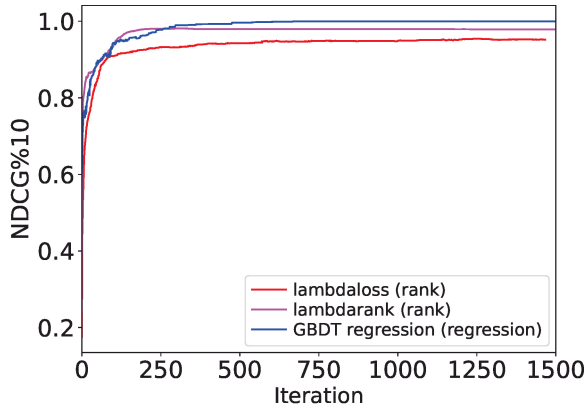


(d) Dataset 2 (simple task), NEDCG

Fig. 2: Prediction results for Dataset 1 and Dataset 2. The evaluation metrics are NDCG and NEDCG of top K samples, and the lines indicate the metrics for increasing values of K. The dashed lines in NEDCG plots represent random guessing (NEDCG = 0).



(a) Dataset 1 (complicated task), NDCG@10



(b) Dataset 2 (simple task), NDCG%10

Fig. 3: NDCG for each iteration of the validation data during the training of the GBDT models. The figure shows (a) the values per iteration of NDCG@10 for Dataset 1 and (b) the values per iteration of NDCG%10 for Dataset 2.

TABLE III: NDCG and NEDCG scores of the top 10 and top 10% samples for Dataset 1 and Dataset 2 for each model. Boldface indicates the value for the method exhibiting the highest prediction accuracy for each dataset. Italics indicate methods judged to have negative values in NEDCG, i.e., worse than random guessing.

	lambdaloss (rank)		lambdarank (rank)		RankSVM (rank)		GBDT regression	
	NDCG	NEDCG	NDCG	NEDCG	NDCG	NEDCG	NDCG	NEDCG
Dataset 1 (top 10, $K = 10$)	**0.593**	**0.286**	0.543	0.212	0.465	0.077	0.404	*−0.056*
Dataset 1 (top 10%, $K = 11$)	**0.600**	**0.600**	0.541	0.541	0.472	0.472	0.413	0.413
Dataset 2 (top 10, $K = 10$)	0.342	0.340	0.278	0.275	**0.368**	**0.366**	0.239	0.236
Dataset 2 (top 10%, $K = 9{,}778$)	0.405	0.364	0.526	0.494	0.310	0.263	**0.559**	**0.529**

V. Discussion

A. Training of the GBDT

Figure 3 shows the convergence of validation scores in the training of the GBDT model. The validation data in Dataset 1 consists of assay data for target proteins that the training data does not contain. Thus, the model overfits the training data as the number of iterations increases. In contrast, the validation data in Dataset 2 are randomly split from the training data. We believe that the similar distribution of the training and validation data is why the scores did not decrease during training. In this experiment, the early stopping parameter was set to 1000 rounds; thus, no concern regarding overfitting was present even with Dataset 1.

B. NEDCG is an intuitive metric

The proposed metric, NEDCG, is effective for complicated tasks such as Dataset 1. In Dataset 1, the NEDCG of the GBDT regression is negative. This score implies that GBDT regression predicts worse than random prediction. If we had used NDCG, we would not have noticed this problem. Using the NEDCG score, we could detect such hazards for the first time. Thus, in a complicated task such as Dataset 1, examining whether the prediction accuracy is sufficient is crucial. However, the difference between the NDCG and NEDCG values was slight because the Dataset 2 results improved substantially from the random predictions. For such datasets, either NEDCG or NDCG provide similar results.

C. Comparison of RankSVM and GBDT

From Dataset 1, the GBDT ranking method was more accurate than RankSVM. The RankSVM method learns ordinal relations for all pairs, whereas the methods using the lambdarank and lambdaloss loss functions aim to optimize NDCG directly. In addition, the GBDT model is more practical than RankSVM for large datasets because the computation time of the RankSVM method increases in the order of the cube of the number of training instances. Thus, the GBDT-based method with lambdarank and lambdaloss outperforms the RankSVM-based method regarding prediction accuracy for LBVS and their applicability to a large dataset.

D. Regression vs. learning-to-rank

This section discusses when to use the learning-to-rank. Although learning-to-rank is superior because it can integrate assay data from different environments to make predictions, as shown in the results section, this approach has the following disadvantages compared to regression.

RankSVM, lambdarank, and lambdaloss train in pairs or listwise, which can be slower than regression. However, in learning-to-rank with GBDT, some techniques reduce the computational complexity, such as branch pruning with `lambdarank_truncation_level`, which is a parameter of LightGBM. Similarly, SPDRank can reduce training time for large datasets by ignoring meaningless order [17]. Therefore, no considerable difference in training time between regression and learning-to-rank methods was observed in most cases.

Thus, the ranking prediction should be used as an alternative when regression prediction cannot achieve practical accuracy, for example, in Dataset 1. The information on activity values obtained in regression is lost in the ranking score. The experimental results with a single assay in Dataset 2 showed little effect of learning-to-rank, and the predictions from regression learning seem to be somewhat reliable. Conversely, in Dataset 1, the regression method was comparable to random prediction, indicating that learning-to-rank was effective for datasets with multiple assays and targets. If the NEDCG is near 0.0, we can quantitatively judge that the model is not predicting adequately. Therefore, we can answer when to use learning-to-rank based on the newly proposed evaluation metric, NEDCG.

VI. Conclusion

We evaluated the ranking prediction performance of the GBDT model using a lambdarank loss function aimed to directly optimize the NDCG to improve the accuracy of VS through learning-to-rank. The comparison between the proposed method with RankSVM and GBDT regression models showed that the proposed method outperformed the other models on a dataset that integrated biochemical assay data from various environments, a case generally considered challenging for regression prediction. We also examined the range of effectiveness of learning-to-rank and found that the effectiveness was not superior to regression methods for all ranking predictions and that learning-to-rank was particularly effective in situations where data integration was essential. In addition, the proposed NEDCG provided a more intuitive evaluation of prediction favorability than the existing NDCG because a value of NEDCG greater than or equal to 0 indicates that the prediction is better than a random prediction.

We focused on the GBDT method using learning-to-rank. However, deep learning models, which have been gaining pop-

ularity in recent years, may further improve the performance of LBVS using learning-to-rank [33]. Although learning-to-rank with deep learning models is not necessarily superior to descriptor-based machine learning methods such as GBDT and RankSVM, an examination of their limitations is required.

Furthermore, learning-to-rank may apply to other tasks in chemoinformatics besides affinity prediction, such as ADMET (absorption, distribution, metabolism, excretion, and toxicity) prediction [34], QSAR (Quantitative Structure-Activity Relationship) [18], and drug-target interaction prediction [35]. For these tasks, ranking predictions may be practical for small datasets for regression prediction or when integrating assay data from various environments.

ACKNOWLEDGMENT

The authors thank Yutaka Akiyama and Keisuke Yanagisawa at the Tokyo Institute of Technology for their constructive discussion and feedback. This work was financially supported by Japan Science and Technology Agency (JST) FOREST (Grant No. JPMJFR216J), JST ACT-X (Grant No. JPMJAX20A3), Japan Society for the Promotion of Science (JSPS) KAKENHI (Grant No. 20H04280), and Japan Agency for Medical Research and Development (AMED) Basis for Supporting Innovative Drug Discovery and Life Science Research (BINDS) (Grant No. JP22ama121026).

REFERENCES

[1] A. Mullard, "New drugs cost US $2.6 billion to develop," *Nat. Rev. Drug Discov.*, vol. 13, p. 877, 2014.

[2] C. M. Song, S. J. Lim, and J. C. Tong, "Recent advances in computer-aided drug design," *Brief. Bioinform.*, vol. 10, no. 5, pp. 579-591, 2009.

[3] X. C. Yan, J. M. Sanders, Y.-D. Gao, M. Tudor, A. M. Haidle, D. J. Klein, A. Converso, C. A. Lesburg, Y. Zang, and H. B. Wood, "Augmenting hit identification by virtual screening techniques in small molecule drug discovery," *J. Chem. Inf. Model.*, vol. 60, no. 9, pp. 4144–4152, 2020.

[4] C. Eleuteri, S. Olla, C. Veroni, R. Umeton, R. Mechelli, S. Romano, M. C. Buscarinu, F. Ferrari, G. Calo, G. Ristori *et al.*, "A staged screening of registered drugs highlights remyelinating drug candidates for clinical trials," *Sci. Rep.*, vol. 7, no. 1, pp. 1–15, 2017.

[5] M. Kandeel and M. Al-Nazawi, "Virtual screening and repurposing of FDA approved drugs against COVID-19 main protease," *Life Sci.*, vol. 251, p. 117627, 2020.

[6] J. Ma, R. P. Sheridan, A. Liaw, G. E. Dahl, and V. Svetnik, "Deep neural nets as a method for quantitative structure–activity relationships," *J. Chem. Inf. Model.*, vol. 55, no. 2, pp. 263–274, 2015.

[7] S. Li, J. Zhou, T. Xu, L. Huang, F. Wang, H. Xiong, W. Huang, D. Dou, and H. Xiong, "Structure-aware interactive graph neural networks for the prediction of protein-ligand binding affinity," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, 2021, pp. 975–985.

[8] N. Schneider, C. Jäckels, C. Andres, and M. C. Hutter, "Gradual in silico filtering for druglike substances," *J. Chem. Inf. Model.*, vol. 48, no. 3, pp. 613–628, 2008.

[9] F. Nigsch, A. Bender, J. L. Jenkins, and J. B. Mitchell, "Ligand-target prediction using Winnow and naive Bayesian algorithms and the implications of overall performance statistics," *J. Chem. Inf. Model.*, vol. 48, no. 12, pp. 2313–2325, 2008.

[10] Y. O. Adeshina, E. J. Deeds, and J. Karanicolas, "Machine learning classification can reduce false positives in structure-based virtual screening," *Proc. Natl. Acad. Sci.*, vol. 117, no. 31, pp. 18477–18488, 2020.

[11] A. M. Wassermann, H. Geppert, and J. Bajorath, "Searching for target-selective compounds using different combinations of multiclass support vector machine ranking methods, kernel functions, and fingerprint descriptors," *J. Chem. Inf. Model.*, vol. 49, no. 3, pp. 582–592, 2009.

[12] T. Joachims, "Optimizing search engines using clickthrough data," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, 2002, pp. 133–142.

[13] S. Agarwal, D. Dugar, and S. Sengupta, "Ranking chemical structures for drug discovery: a new machine learning approach," *J. Chem. Inf. Model.*, vol. 50, no. 5, pp. 716–731, 2010.

[14] F. Rathke, K. Hansen, U. Brefeld, and K.-R. Müller, "StructRank: a new approach for ligand-based virtual screening," *J. Chem. Inf. Model.*, vol. 51, no. 1, pp. 83–92, 2011.

[15] W. Zhang, L. Ji, Y. Chen, K. Tang, H. Wang, R. Zhu, W. Jia, Z. Cao, and Q. Liu, "When drug discovery meets web search: learning to rank for ligand-based virtual screening," *J. Cheminform.*, vol. 7, no. 1, p. 5, 2015.

[16] S. D. Suzuki, M. Ohue, and Y. Akiyama, "PKRank: a novel learning-to-rank method for ligand-based virtual screening using pairwise kernel and ranksvm," *Artif. Life Robot.*, vol. 23, no. 2, pp. 205–212, 2018.

[17] M. Ohue, S. D. Suzuki, and Y. Akiyama, "Learning-to-rank technique based on ignoring meaningless ranking orders between compounds," *J. Mol. Graph. Model.*, vol. 92, pp. 192–200, 2019.

[18] K. Matsumoto, T. Miyao, and K. Funatsu, "Ranking-Oriented Quantitative Structure–Activity Relationship Modeling Combined with Assay-Wise Data Integration," *ACS Omega*, vol. 6, no. 18, pp. 11964–11973, 2021.

[19] C. J. Burges, "From RankNet to LambdaRank to LambdaMART: An Overview," Tech. Rep. MSR-TR-2010-82, 2010.

[20] K. Järvelin and J. Kekäläinen, "IR evaluation methods for retrieving highly relevant documents," in *ACM SIGIR Forum*, vol. 51, no. 2, 2017, pp. 243–250.

[21] K. Järvelin and J. Kekäläinen, "Cumulated gain-based evaluation of IR techniques," *ACM Trans. Inf. Syst.*, vol. 20, no. 4, pp. 422–446, 2002.

[22] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender, "Learning to rank using gradient descent," in *Proc. 22nd Int. Conf. Mach. Learn*, 2005, pp. 89–96.

[23] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, 2016, pp. 785–794.

[24] 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," *Adv. Neural Inf. Process. Syst.*, vol. 30, pp. 3146–3154, 2017.

[25] C. Burges, R. Ragno, and Q. Le, "Learning to rank with nonsmooth cost functions," *Adv. Neural Inf. Process. Syst.*, vol. 19, pp. 193–200, 2006.

[26] X. Wang, C. Li, N. Golbandi, M. Bendersky, and M. Najork, "The lambdaloss framework for ranking metric optimization," in *Proc. 27th ACM Int. Conf. Inf. Knowl. Manag.*, 2018, pp. 1313–1322.

[27] D. Mendez, A. Gaulton, A. P. Bento, J. Chambers, M. De Veij, E. Félix, M. P. Magariños, J. F. Mosquera, P. Mutowo, M. Nowotka, M. Gordillo-Marañón, F. Hunter, L. Junco, G. Mugumbate, M. Rodriguez-Lopez, F. Atkinson, N. Bosc, C. J. Radoux, A. Segura-Cabrera, A. Hersey, and A. R. Leach, "ChEMBL: towards direct deposition of bioassay data," *Nucleic Acids Res.*, vol. 47, no. D1, pp. D930-D940, 2018.

[28] Y. Wang, T. Suzek, J. Zhang, J. Wang, S. He, T. Cheng, B. A. Shoemaker, A. Gindulyte, and S. H. Bryant, "PubChem bioassay: 2014 update," *Nucleic Acids Res.*, vol. 42, no. D1, pp. D1075–D1082, 2014.

[29] H. Moriwaki, Y.-S. Tian, N. Kawashita, and T. Takagi, "Mordred: a molecular descriptor calculator," *J. Cheminform.*, vol. 10, no. 1, p. 4, 2018.

[30] D. Rogers and M. Hahn, "Extended-connectivity fingerprints," *J. Chem. Inf. Model.*, vol. 50, no. 5, pp. 742–754, 2010.

[31] L. Gregory, "RDKit: open-source cheminformatics," http://www.rdkit.org.

[32] T.-M. Kuo, C.-P. Lee, and C.-J. Lin, "Large-scale kernel RankSVM," in *Proc. 2014 SIAM Int. Conf. Data Min.*, 2014, pp. 812–820.

[33] D. Jiang, Z. Wu, C.-Y. Hsieh, G. Chen, B. Liao, Z. Wang, C. Shen, D. Cao, J. Wu, and T. Hou, "Could graph neural networks learn better molecular representation for drug discovery? A comparison study of descriptor-based and graph-based models," *J. Cheminform.*, vol. 13, no. 1, p. 12, 2021.

[34] J. Li, K. Yanagisawa, Y. Yoshikawa, M. Ohue, and Y. Akiyama, "Plasma protein binding prediction focusing on residue-level features and circularity of cyclic peptides by deep learning," *Bioinformatics*, vol. 38, no. 4, pp. 1110-1117, 2022.

[35] X. Ru, X. Ye, T. Sakurai, and Q. Zou, "NerLTR-DTA: drug – target binding affinity prediction based on neighbor relationship and learning to rank," *Bioinformatics*, vol. 38, no. 7, pp. 1964-1971, 2022.