Quantifying Overfitting Potential in Drug Binding Datasets*

Brian Davis[1], Kevin Mcloughlin[2], Jonathan Allen[2], and Sally R Ellingson[1]

[1] University of Kentucky Markey Cancer Center
sel228@uky.edu
[2] Lawrence Livermore National Laboratory

Abstract. In this paper, we investigate potential biases in datasets used to make drug binding predictions using machine learning. We investigate a recently published metric called the Asymmetric Validation Embedding (AVE) bias which is used to quantify this bias and detect overfitting. We compare it to a slightly revised version and introduce a new weighted metric. We find that the new metrics allow to quantify overfitting while not overly limiting training data and produce models with greater predictive value.

Keywords: Drug Discovery · Machine Learning · Data Overfitting.

1 Introduction

Protein-ligand interactions are important to most processes in the human body, and therefore to regulating disease via drugs. There are an estimated 20,000 different human protein-coding genes[3], and 10^{60} small molecules in the chemical universe[10]. Clearly, exploring all possible protein–drug pairs is not experimentally feasible. Drug discovery programs need accurate computational methods to predict protein–drug binding, and advances in machine learning have improved the accuracy of these predictions used in early stages of drug discovery. For a review of some current trends in making drug binding predictions using machine learning and available datasets see this current review [4].

The primary goal in protein–ligand binding modeling is to produce models that are capable of making accurate predictions on novel protein–drug pairs. Consequently, performance metrics need to reflect expected performance on novel data. This effort is frustrated by benchmark datasets that are not well-sampled from chemical space, so that novel pairs may be relatively far from data available to the modeler. Area under the curve (AUC) scores for various curves are often provided to support suitability of a machine learning model for use in drug binding prediction— in this paper we focus on the Precision–Recall curve and its associated AUC (PR-AUC). Care must be taken in interpreting

* Supported by Lawrence Livermore National Laboratory and the University of Kentucky Markey Cancer Center.

[3] with about one-eighth of the exome containing observable genetic variations [8]

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50420-5_44

performance metrics like the PR-AUC, as laboratory experiments reveal that inferring real-world performance from AUC alone is overly-optimistic. This issue of generalizability is common in machine learning applications, but is particularly relevant in settings with insufficient and non-uniformly distributed data, as is the case with drug binding data.

The phenomenon of high performance metrics for low quality models is called overfitting, and is typically combated by using different data for the processes of model training and validation. If the validation data and real-world data for the model application are both distinct from the training data, then we expect the performance metrics on the validation data to be representative of the real-world performance of the model. A common way of splitting the available data into training and validation sets is to select a training ratio and randomly assign that proportion of the data to the training set.

A developing framework to account for overfitting is based on the assumption that the Nearest Neighbor (NN) model has poor generalizability. In the NN model, the test data is classified based on the classification of its nearest neighbor in the training data. Therefore, NN basically memorizes the training data and does not generalize to anything not close to it. Within the context of reporting "fair" performance metrics, this working assumption of poor generalizability of NN models suggests several possibilities for more informative metrics, including:

1. reporting the PR-AUC for a model produced from a training/validation split on which the Nearest Neighbor model has poor performance, and
2. designing a metric which weights each validation molecule according to its relative distance to the binding classes in the training set.

We describe implementations of each of these approaches in this paper. For the first approach, we discuss the efforts presented in the Atomwise paper[14] to produce training/validation splits that are challenging for NN models, hereafter referred to as the Atomwise algorithm. We also describe two variations of the Atomwise algorithm: ukySplit–AVE and ukySplit–VE. As distinct from optimization, we introduce a weighting scheme ω designed to address the second approach, and discuss the consequences of using an ω-weighted PR-AUC versus the traditional PR-AUC with a training/validation split produced by the ukySplit–AVE algorithm.

To summarize the problem at hand, there are too many combinations of potential therapies and drug targets to understand them by experiment alone; simulations still suffer from inaccuracies and high computational costs; with the wealth of biomedical data available, machine learning is an attractive option; current models suffer from data overfitting, where feature sets are too large and patterns can be observed that do not generalize to new data; the current project discusses ways of quantifying and better understanding the potential for overfitting without limiting the predictability of models.

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50420-5_44

1.1 Current Bias Quantification Methods

For a more articulated description of the original spatial features that inspired this work, including an evaluation of several benchmark datasets with figures to help conceptualize the ideas please see the recent review work [4].

Datasets with a metric feature space can be evaluated using spatial statistics [12] to quantify the dataset topology and better understand potential biases. Of particular interest in the area of drug–binding model generalization are the "nearest neighbor function" G(t) and the "empty space function" F(t). G(t) is the proportion of active compounds for whom the distance to the nearest active neighbor is less than t. F(t) is the proportion of decoy compounds for whom the distance to the nearest active neighbor is less than t. Letting $\sum$G and $\sum$F denote the sum of the values of G and F over all thresholds t, it is reported that large values of $\sum$G indicate a high level of self-similarity and that small values of $\sum$F indicate a high degree of separation. The difference of $\sum$G and $\sum$F gives a quick and interpretable summary of a dataset's spatial distribution, with negative values indicating clumping, near-zero values indicating a random-like distribution, and positive values indicating larger active-to-active distance than decoy-to-active. These spatial statistics were used to develop the Maximum Unbiased Validation (MUV) dataset, with the goal of addressing the reported association of dataset clumping with overly–optimistic virtual screening results [12, 11].

Wallach et al. [14] extended the MUV metric, and used it to quantify the spatial distribution of actives and decoys among the training and validation sets. For two subsets V and T of a metric data set with distance function d, define, for each v in V, the function $I_t(v, T)$ to be equal to one if $\min_{w \in T}\{d(v, w)\} < t$ and zero otherwise. For a fixed value of n, define the function $H_{(V,T)}$ by

$$H_{(V,T)} = \frac{1}{n+1} \cdot \frac{1}{|V|} \sum_{v \in V} \left(\sum_{i=0}^{n} I_{i/n}(v, T) \right). \tag{1}$$

Then the Asymmetric Validation Embedding (AVE) bias is defined to be the quantity

$$B(V_A, V_I, T_A, T_I) = H_{(V_A,T_A)} - H_{(V_A,T_I)} + H_{(V_I,T_I)} - H_{(V_I,T_A)}, \tag{2}$$

where the value of n is taken to be 100, and where V_A and V_I are the validation actives and inactives (decoys), respectively, and similarly T_A and T_I are the training actives and inactives. For convenience, we abbreviate $H(V_a, T_a) - H(V_a, T_i)$ and $H(V_i, T_i) - H(V_i, T_a)$ as $(AA - AI)$ and $(II - IA)$, respectively. They are intended to be a quantification of the "clumping" of the active and decoy sets. If the term $(AA - AI)$ is negative, it suggests that, in the aggregate, the validation actives are closer to training decoys than to training actives, with the consequence that the active set is expected to be challenging to classify. If the sum of $(AA - AI)$ and $(II - IA)$ (the AVE bias) is close to zero, it is expected that the data set is "fair", in that it does not allow for easy classification due

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50420-5_44

to clumping. The authors also provide an AVE bias minimization algorithm. It is a genetic algorithm with breeding operations: merge, add molecule, remove molecule, and swap subset. The algorithm first generates initial subsets through random sampling, measures the bias, and selects subsets with low biases for breeding. The algorithm repeats bias scoring, redundancy removal, and breeding until termination based on minimal bias or maximum iterations.

In their paper, Wallach et al. observe that AUC scores[4] and AVE bias scores are positively correlated for several benchmark data sets, implying that model performance is sensitive to the training/validation split.

In this paper, we present an efficient algorithm for minimizing the AVE bias of training/validation splits. We introduce a variation on the AVE bias, which we call the VE score, and describe its advantages in the context of optimization. We investigate the efficacy of minimizing these metrics for training/validation splits, and conclude by proposing a weighted performance metric as an alternative to the practice of optimizing training/validation splits.

2 Methods

2.1 Dataset

Dekois 2 [2] provides 81 benchmark datasets: 80 with unique proteins, and one with separate datasets for two different known binding pockets in the same protein. The active sets are extracted from BindingDB [6]. Weak binders are excluded, and 40 distinct actives are selected by clustering Morgan fingerprints by Tanimoto similarity. Three datasets are extended by selecting up to 5 actives from each structurally diverse cluster. The decoy set is generated using ZINC [7] and property matched to the actives based on molecular weight, octanol-water partition coefficient (logP), hydrogen bond acceptors and donors, number of rotatable bonds, positive charge, negative charge, and aromatic rings. Possible latent actives in the decoy set are removed using a score based on the Morgan fingerprint and the size of the matching substructures. Any decoy that contained a complete active structure as a substructure is also removed.

2.2 Bias metrics

Throughout this paper, the term fingerprint refers to the 2048-bit Extended Connectivity Fingerprint (ECFP6) of a molecule as computed by the Python package RDKit [1]. For simplicity, we define

$$d(v, T) := \min_{t \in T} \{d(v, t)\}$$

and

$$\Gamma(v, T) := \frac{\lfloor n \cdot d(v, T) \rfloor}{n + 1},$$

[4] They report ROC-AUC scores, as opposed to PR-AUC scores.

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50420-5_44

where $d(v,t)$ is the Tanimoto distance between the fingerprints of the molecules v and t. We compute the AVE bias via the expression

$$\text{mean}_{v\in V_A}\{\Gamma(v,T_I)-\Gamma(v,T_A)\}+\text{mean}_{v\in V_I}\{\Gamma(v,T_A)-\Gamma(v,T_I)\}, \quad (3)$$

where V_A and V_I are the validation actives and inactives (decoys), respectively, and similarly T_A and T_I are the training actives and inactives. For a derivation of the equivalence of this expression and Expression (2), see the Appendix. Since

$$|d(v,T)-\Gamma(v,T)|<\frac{1}{n+1},$$

for large values of n Expression (3) (and hence the AVE bias) is an approximation of

$$\text{mean}_{v\in V_A}\{d(v,T_I)-d(v,T_A)\}+\text{mean}_{v\in V_I}\{d(v,T_A)-d(v,T_I)\}. \quad (4)$$

We now introduce the VE score, a close relative of the AVE bias:

$$\sqrt{\text{mean}^2_{v\in V_A}\{d(v,T_I)-d(v,T_A)\}+\text{mean}^2_{v\in V_I}\{d(v,T_A)-d(v,T_I)\}}. \quad (5)$$

While the raw ingredients of the AVE bias and the VE score are the same, they are qualitatively different, in particular as the VE score is never negative.

We generate a random training/validation split for each Dekois target and evaluate Expressions (2) through (5) 1,000 times with a single thread on an AMD Ryzen 7 2700x eight-core processor. We compare the mean computation times, as well as the computed values.

2.3 Split Optimization

We implement two custom genetic optimizers, ukySplit-AVE and ukySplit-VE, using the open source DEAP [5] framework. Both ukySplit-AVE and ukySplit-VE optimizers use parameters as described in Table 1. The parameters were chosen after grid-searching for minimum mean-time-to-debias on a sample of the Dekois targets.

Parameter Name	Meaning	Value
POPSIZE	Size of the population	500
NUMGENS	Number of generations in the optimization	2000
TOURNSIZE	Tournament Size	4
CXPB	Probability of mating pairs	0.175
MUTPB	Probability of mutating individuals	0.4
INDPB	Probability of mutating bit of individual	0.005

Table 1. Evolutionary algorithm parameters

The optimizer populations consisted of training/validation splits, and the objective functions were given by Expressions (3) and (5), respectively, for *valid* splits, and equal to 2.0 otherwise. We say that a split is valid if

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50420-5_44

1. the validation set contains at least one active and one decoy molecule,
2. the active/decoy balance in the validation set is within 5% of that in the total dataset,
3. the ratio of training/validation set sizes is $80 \pm 1\%$.

2.4 Modeling

Using scikit-learn [9], we train a random forest classifier ($n\_estimators = 100$) with stratified 5-fold cross-validation and compute the mean PR-AUC for each target of the Dekois data set. We use fingerprints as features, and take the probability of the active class as the output of the model. For each of the folds, we evaluate the Expressions (3) and (5), and report Pearson correlation coefficients with the PR-AUC.

For the training/validation splits produced by an optimizer, we compute PR-AUC of a random forest model and evaluate Expression (3) or (5) as applicable.

2.5 Nearest Neighbor similarity

An assumption this paper builds upon is that the Nearest Neighbor model does not generalize well since it memorizes the training data. Good metrics come from a Nearest Neighbor model that is only tested using data points very similar to data points in the training data and having the same label. Therefore, we use a similarity measure to the Nearest Neighbor model to show the potential of a model to not generalize.

We gather the binary predictions made by the Nearest Neighbor model, which predicts the class of a validation molecule to be the same as its nearest neighbor (using the metric d) in the training set. Considering the NN predictions as a bit string, we can compare it with the prediction bit string of any other model m using the Tanimoto similarity T:

$$T(NN, m) = \frac{\sum (NN \wedge m)}{\sum (NN \vee m)},$$

with bitwise operations $\wedge$ (and) and $\vee$ (or) and sums over all predictions for the validation set. We take the maximum Tanimoto similarity over all thresholds η for each of the validation folds, and report the mean.

2.6 Weighted PR-AUC

The weighted metric described here gives less of a contribution to the model's performance metric when a tested data point is very similar to a training data point.

For a given model, let TP, TN, FP, and FN be the collections of molecules for which the model predictions are true positive, true negative, false positive, and false negative, respectively. The metrics precision and recall may be easily

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50420-5_44

generalized by assigning a weight $\omega(v)$ to each molecule v, and letting the ω–weighted precision be given by

$$\frac{\sum_{v\in \mathrm{TP}} \omega(v)}{\sum_{v\in \mathrm{TP}} \omega(v) + \sum_{v\in \mathrm{FP}} \omega(v)}$$

and the ω–weighted recall be given by

$$\frac{\sum_{v\in \mathrm{TP}} \omega(v)}{\sum_{v\in \mathrm{TP}} \omega(v) + \sum_{v\in \mathrm{FN}} \omega(v)}.$$

Setting the weight $\omega(v)$ equal to 1 for all molecules v, we recover the standard definitions of precision and recall.

Inspired by Expression (4), we define the ratio $\gamma(v)$ by

$$\gamma(v) = \begin{cases} \frac{d(v,T_A)}{d(v,T_I)} & \text{if } v \text{ is active,} \\ \\ \frac{d(v,T_I)}{d(v,T_A)} & \text{if } v \text{ is decoy.} \end{cases}$$

When we refer to the weighted PR-AUC in this paper we use the weight ω given by the cumulative distribution function of γ over the validation set for the target protein. Note that the weights ω are between zero and one, and that the weighting de-emphasizes molecules that are closer to training molecules of the same binding class than to training molecules of the opposite class. Thus the larger the contribution of a molecule to the AVE bias, the lower its weight. For further description of the ω–weighted PR-AUC, see the Appendix of [3].

2.7 Generalizability

Inspired by recent work presented on the so-called "far AUC"[13], we attempt to measure the ability of a drug-binding model to generalize. We randomly split the data set for each target 80/20 (preserving the class balance), then remove any molecules in the 80% set that has a distance less than 0.4 from the 20% set. We reserve the 20% set to serve as a proxy for novel data "far" from the training data. We then treat the remainder of the 80% as a data set, running the same analysis as described in the earlier subsections: computing the weighted and unweighted PR-AUC of a random forest trained on random splits, as well as the PR-AUC of random forest models trained on ukySplit-AVE and ukySplit-VE optimized splits.

3 Results

3.1 Computational Efficiency

A naive implementation of Equation (2) required a mean computation time over all Dekois targets of 7.14 ms, while an implementation of Equation (3) had a

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50420-5_44

Expression	Mean Computation Time	Relative Speedup
(2)	7.14 ms	1
(3)	0.99 ms	7.2
(4)	0.31 ms	23.4
(5)	0.31 ms	23.1

Table 2. Computational Efficiency

mean computation time of 0.99 ms. The mean computation times for Expressions (4) and (5) were both approximately 0.31 ms.

Evaluations of Expressions (2) through (5) are plotted in Figure 1. The absolute differences between the computed value of Expression (2) and Expressions (3) and (4) are summarized in Table 3. It is not meaningful to compare the evaluations of Expressions (2) and (5) in this way, as they measure different, though related, quantities.

Expression	Mean Abs Difference	Max Abs Difference
(3)	3.1×10^{-4}	4.1×10^{-3}
(4)	9.9×10^{-3}	2.3×10^{-2}

Table 3. Comparison with Expression (2) over Dekois targets

For reference, the AVE paper considered a score of 2×10^{-2} to be "bias free".

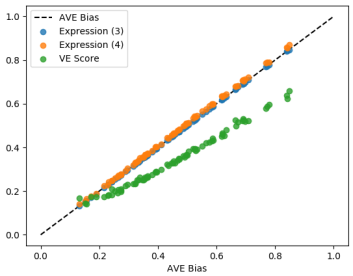

Fig. 1. Comparison of evaluations over Dekois targets

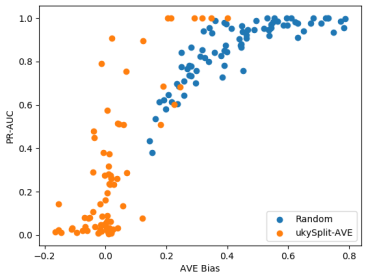

Fig. 2. Mean Split AVE Bias vs. Model PR-AUC.

3.2 Split Bias and Model Performance

In Figure 2, we plot the mean PR-AUC against the mean AVE bias for 5-fold cross validation on each Dekois target. The Pearson correlation coefficient

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50420-5_44

between mean AVE bias and mean PR-AUC is computed to be 0.80, which is comparable in strength to the correlation reported in the AVE paper for other benchmark datasets. We also plot the AVE bias against the mean PR-AUC for each target after optimization by ukySplit–AVE. Note that, although the optimizer was run with a stopping criterion of 0.02, it is possible for the minimum AVE bias to jump from greater than 0.02 to a negative number (as low as -0.2) in one generation.

We order the target proteins by AVE bias, and plot the two components, AA-AI and II-IA, after optimization by ukySplit–AVE in Figure 3.

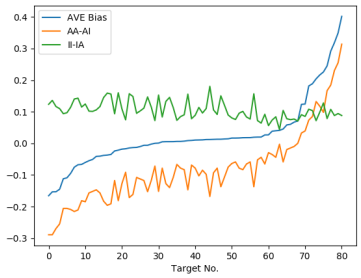

Fig. 3. The two components of the AVE Bias score.

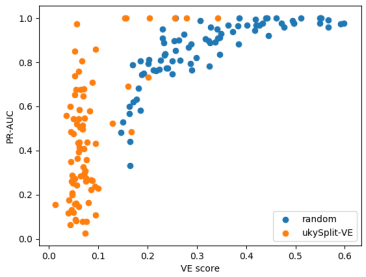

Fig. 4. Mean VE Score vs. Model PR-AUC.

3.3 Optimization by ukySplit–VE

Figure 4 plots the mean VE score against the mean PR-AUC across each fold of the cross-validation split for each target before and after optimization with ukySplit–VE (minimizing VE score as opposed to AVE bias). Figure 5 plots the score components associated to the active and decoy validation molecules after optimizing VE score (for comparison with Figure 3).

3.4 Weighted Performance

Figure 6 plots the ω–weighted PR-AUC against the mean AVE bias over each fold for each target. Recall that the models and predictions are the same as those represented in Figure 2 with label "random", but that the contributions of each validation molecule to the precision and recall are now weighted by the weight function ω.

3.5 Nearest Neighbor similarity

Figure 7 plots the NN- similarity of a random forest model trained on splits produced randomly, by ukySplit-AVE, and by ukySplit-VE. The mean NN-similarities were 0.997, 0.971, and 0.940, respectively.

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50420-5_44

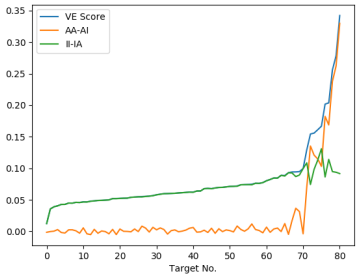

Fig. 5. The two components of the VE Score.

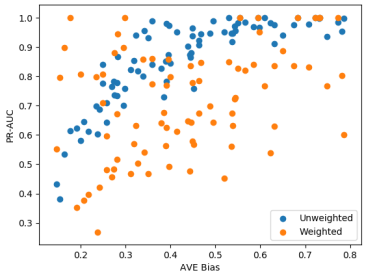

Fig. 6. Mean Split AVE Bias vs. Model Weighted PR-AUC.

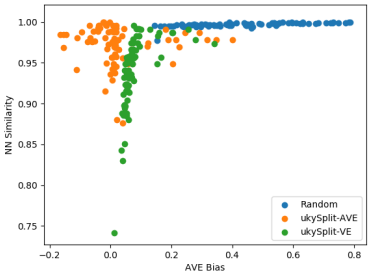

Fig. 7. NN similarity of a random forest model trained on various splits.

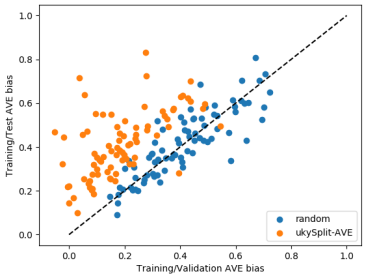

Fig. 8. AVE bias of training/validation and training/test splits.

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50420-5_44

3.6 Model Performance on Distant Data

After reserving 20% of each target's data for the test set, approximately 3% of the remainder was found to be within the 0.4 buffer distance, and was removed before splitting into training and validation sets. Similarly to Sundar and Colwell [13], we find that de-biasing training/validation splits does not lead to increased performance on "distant" test sets: the mean ratio of test PR-AUC before and after split optimization by ukySplit-AVE was 1.010 (1.018 for ukySplit-VE).

Figure 8 plots the AVE bias on the training/test split against the AVE bias on the training/validation split (letting the test set play the role of the validation set in the AVE bias definition). Figure 9 shows the validation and test PR-AUC of a model trained with a training set produced randomly, by ukySplit-AVE, and by ukySplit-VE.

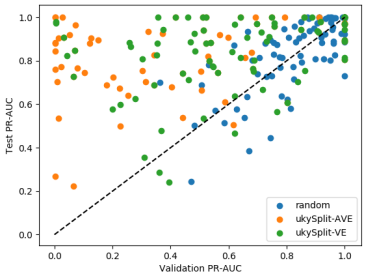

Fig. 9. Model performance on validation vs. test set of a model trained on various splits.

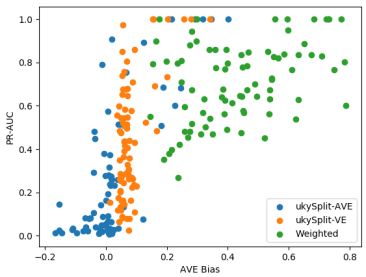

Fig. 10. Comparison of ukySplit-AVE, ukySplit-VE, and weighting methods.

4 Discussion

4.1 Computing Bias

As presented in Table 2, refactoring Expression (2) into Expression (3) yielded speedups of 7x, and the additional use of exact, rather than approximated, values yielded a speedup of roughly 23x for Expression (4). While Expressions (2) and (3) are mathematically equivalent, in practice they yield slightly different results due to machine precision. In the aggregate over the Dekois dataset, the difference is negligible relative to the established definition of "de-biased", as described in Table 3. Expressions (2) and (4) are *not* mathematically equivalent. In light of the equivalence of Expressions (2) and (3), it is clear that AVE bias (Expression (2)) is an *approximation* of Expression (4). Their difference, though slight, is properly interpreted as approximation error in the AVE bias.

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50420-5_44

4.2 Model effects of debiasing

Figures 2 and 4 demonstrate that the process of minimizing bias in the training / validation split risks training a model with little or no predictive value. The expected recall (and precision) for a random guessing model is equal to the balance of active molecules, which for the Dekois dataset is 3%. Of the 81 Dekois targets, 21 (about 26%) had models with below random PR-AUC when trained and validated on a split produced by ukySplit–AVE. This may be understood by considering Figure 3, which shows that the AVE bias is primarily an indication of the spatial distribution of the (minority) active class in the validation set, with low AVE bias associated with active validation molecules that are closer to training decoys than to training actives. Models trained on such splits are therefore prone to misclassify validation actives, and hence have a low PR-AUC. This phenomenon is less pronounced when splits are optimized for VE score (ukySplit–VE), as it does not allow terms to "cancel out", and so does not incentivize pathological distributions of validation actives. It can be seen when comparing Figures 3 and 5. When using the AVE bias score (in Figure 3), AA-AI can get "stuck" at a negative value and then II-IA tends towards the absolute value of AA-AI to result in an overall bias score near zero. When using the VE score (in Figure 5), this does not happen. Since AA-AI can never be negative, II-IA will not try to cancel it out. Only one Dekois target had worse-than-random performance for a model trained on a split optimized for VE score, and while the mean PR-AUC over models trained with ukySplit–AVE splits was 0.26, the mean PR-AUC for models trained on ukySplit–VE splits was 0.44. Since one of the assumptions built upon in this paper is that the Nearest Neighbor model does not generalize well, we use a measure of similarity to the Nearest Neighbor model as a measure of potential to not generalize well. It can be seen in Figure 7 that models built on random data can be assumed to not generalize well, that models built on data splits using the AVE bias may sometimes do better but it does not completely alleviate the problem, and that models built on data splits using the VE bias do a better job at diverging from a Nearest Neighbor model.

Therefore, VE may be a better score to debias datasets because datasets debiased with VE are less similar to the NN model and produce a higher PR-AUC.

4.3 Weighted PR-AUC

As described in the Introduction, a weighted metric represents an alternative way of taking into account the assumption of poor generalizability of Nearest Neighbor models. While bias optimization creates training/validation splits that are more challenging for Nearest Neighbor models, they simultaneously result in low quality models, even when using powerful methods like random forests. In Figure 5, when using an unweighted metric with random data the expected trend of higher bias scores coming from models with better performance metrics can be seen and it can be assumed that this is from data overfitting. However, the trend is not present (or much less pronounced) when using the weighted metric.

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50420-5_44

The assumption here is that the weighted metric gives a better representation of the generalizability of the model without having to limit the training data. The weighted metric ω-PR-AUC discounts the potentially inflated performance of models without degrading the models themselves (see Figure 10). It is worth noting, as well, that the computational expense of computing the weighting ω is negligible compared with the intensive work performed in optimizing training / validation splits.

The weighted PR-AUC may be used to infer that the standard PR-AUC is inflated due to the spatial distribution of the validation set. In particular, if two trained models have the same performance, the one with the higher weighted performance may be expected to have better generalizability.

4.4 Test Performance

Figure 8 shows that minimizing the AVE bias on the training/validation split does not minimize the AVE bias on the training/test split. Figure 9 demonstrates that even when a split results in a trained model with very low validation PR-AUC, the model still performs fairly well on the test data.

5 Conclusions

In this paper an existing bias metric was evaluated that quantities potential for data overfitting and can be used to optimize splits in order to build models with test data that give a better indication of how generalizable a model may be. An improvement was made to the score by not allowing one of the terms to be negative which was leading to problems during data split optimizations. However, the biggest contribution is the introduction of using a weighted metric instead of optimized data splits. This allows for the training data to not be limited which leads to models with no predictive power while not inflating performance metrics. This will hopefully lead to models that are more predictive on novel real world test data with performance metrics that better represent that potential. Developers of machine learning models for virtual high-throughput screening will have to contend with issues of overfitting as long as drug binding data is scarce. We propose the use of weighted performance metrics as a less computation-intensive alternative to split optimization. If the weighted and un-weighted metrics diverge, it can be concluded that the good performance of a model is concentrated at data points on which a nearest-neighbor model is sufficient.

Future work includes combining a protein distance with the drug distance to represent protein-drug pairs in a dataset. This is to evaluate large datasets to make multi-protein prediction models.

References

1. Rdkit, open-source cheminformatics. http://www.rdkit.org

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50420-5_44

2. Bauer, M.R., Ibrahim, T.M., Vogel, S.M., Boeckler, F.M.: Evaluation and optimization of virtual screening workflows with dekois 2.0–a public library of challenging docking benchmark sets. Journal of chemical information and modeling **53**(6), 1447–1462 (2013)
3. Davis, B., Mcloughlin, K., Allen, J., Ellingson, S.: Split optimization for protein/ligand binding models. arXiv preprint arXiv:2001.03207 (2020)
4. Ellingson, S.R., Davis, B., Allen, J.: Machine learning and ligand binding predictions: A review of data, methods, and obstacles. Biochimica et Biophysica Acta (BBA)-General Subjects **1864**(6), 129545 (2020)
5. Fortin, F.A., De Rainville, F.M., Gardner, M.A., Parizeau, M., Gagné, C.: DEAP: Evolutionary algorithms made easy. Journal of Machine Learning Research **13**, 2171–2175 (jul 2012)
6. Gilson, M.K., Liu, T., Baitaluk, M., Nicola, G., Hwang, L., Chong, J.: Bindingdb in 2015: a public database for medicinal chemistry, computational chemistry and systems pharmacology. Nucleic acids research **44**(D1), D1045–D1053 (2015)
7. Irwin, J.J., Sterling, T., Mysinger, M.M., Bolstad, E.S., Coleman, R.G.: Zinc: a free tool to discover chemistry for biology. Journal of chemical information and modeling **52**(7), 1757–1768 (2012)
8. Lek, M., Karczewski, K.J., Minikel, E.V., Samocha, K.E., Banks, E., Fennell, T., O'Donnell-Luria, A.H., Ware, J.S., Hill, A.J., Cummings, B.B.: Analysis of protein-coding genetic variation in 60,706 humans. Nature **536**(7616), 285 (2016)
9. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research **12**, 2825–2830 (2011)
10. Reymond, J.L., Awale, M.: Exploring chemical space for drug discovery using the chemical universe database. ACS chemical neuroscience **3**(9), 649–657 (2012)
11. Rohrer, S.G., Baumann, K.: Impact of benchmark data set topology on the validation of virtual screening methods: exploration and quantification by spatial statistics. Journal of chemical information and modeling **48**(4), 704–718 (2008)
12. Rohrer, S.G., Baumann, K.: Maximum unbiased validation (muv) data sets for virtual screening based on pubchem bioactivity data. Journal of chemical information and modeling **49**(2), 169–184 (2009)
13. Sundar, V., Colwell, L.: Debiasing algorithms for protein ligand binding data do not improve generalisation (2019)
14. Wallach, I., Heifets, A.: Most ligand-based classification benchmarks reward memorization rather than generalization. Journal of chemical information and modeling **58**(5), 916–932 (2018)

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50420-5_44