Application of machine learning to predict unbound drug bioavailability in the brain

J. Francisco Morales[1], M. Esperanza Ruiz[1], Robert E. Stratford[2] and Alan Talevi[1*]

[1] Laboratory of Research and Development of Bioactives (LIDeB), CONICET, Departamento de Ciencias Biológicas, Facultad de Ciencias Exactas, Universidad Nacional de La Plata (UNLP). La Plata, Argentina

[2] Division of Clinical Pharmacology, Indiana University School of Medicine, Research II, Suite 480 950 W. Walnut St Indianapolis, IN, USA

Table Contents

Machine Learning Algorithms description

Support Vector Machine (SVM). SVM is a kernel based approach. Briefly, SVM maps the data into a high-dimensional hyperplane, using a kernel function that is typically linear, radial, or polynomial (1). The SVM seeks to find an optimal separation between two classes (*e.g.* inhibitors and non-inhibitors), such that each in their entirety lie on opposite sides of a separating hyperplane. Thus, SVM minimizes the empirical classification error and maximizes the geometric margin. This margin is defined as the distance from the separating hyperplane to its nearest sample. The hyperplane that defines such a margin is called support hyper planes, and the data points that lie on these hyper planes are called support vectors. Thus, SVM is also known as a maximum margin classifier.

Gradient Boosting Modeling (GBM). The GBM method differs from bagging methods because the base learner trees are trained and combined sequentially (1,2). The principle idea behind this algorithm is to generate models by computing a sequence of trees, in which each successive tree is built from the prediction residuals of the preceding tree. A simple (best) partitioning of the data is determined at each step in the boosting tree algorithm, and the deviations of the observed values from the respective residuals for each partition are computed. Given the preceding sequence of trees, the next tree will then be fitted to the residuals in order to find another partition that will further reduce the residual (error) variance for the data.

k-Nearest Neighbors (kNN). The kNN is a non-parametric method that classifies samples based on a similarity measure (1). In this method, a sample could be classified by a majority vote of its neighbors, with the sample investigated being assigned to the class most common amongst its k nearest neighbors measured by a distance function. This distance is usually taken to be the euclidean distance. If k = 5, then the case is simply assigned to the class of its five nearest neighbors in a feature space. The samples, which in chemical applications are typically compounds, are described as position vectors in the feature space, which is usually of high dimensionality.

Classificatory Partial Least Squares (cPLS). cPLS is a parametric method based on the PLS model in which the dependent variable is chosen to represent the class membership (3). First a classical PLS model is built with a training set. In PLS, the number of variables is reduced using PCA by creating new latent variables which maximize the covariance between the original variables and the response. Using the optimal number of latent variables, to build a linear regression model should provide the best predictive model. For an unknown sample, the predicted value obtained with the cPLS model is regularly distributed around 0 or 1.

Random Forest (RF). The RF algorithm is a tree bagging method that creates a large collection of decorrelated decision trees, and the final prediction is defined by majority voting from an ensemble of decision trees (1,4). In each tree, one-third of the training set is randomly extracted, while the remaining two-third of the training set is used for model building. When these decision trees are built, each time a split in a tree is considered, a random sample of predictors is chosen as split candidates from the full set of predictors. Each tree is grown to the largest possible extent without pruning. Last, the trained forest is then used to predict the remaining one-third of the observations not used to fit the model and calculate the out-of-bag (OOB) error. The predicted classification values are defined by majority voting for one of the classes.

Deep Learning (DL). DL algorithm is a network composed of units called artificial neurons (5). Each connection between neurons can transmit a signal to another neuron. All neurons have multiple inputs and one output. The receiving neuron can process the signals and then signal downstream neurons connected to it. Each input is associated with a weight that varies as learning proceeds, which can increase or decrease the strength of the signal that it sends downstream. Typically, neurons are organized in layers. Normally, there are three types of layers in DL: (1) the input layer (*i.e.,* the bottom layer), where the descriptors of a molecule are entered. (2) the hidden (middle) layers; the word “deep” in DL implies more than one hidden layer. (3) the output layer (*i.e.,* the top layer) where predictions are generated.

Extreme gradient boosting (XGBOOST). XGBOOST algorithm is a meta-algorithm to construct a strong ensemble learner from weak learners such as decision trees (6,7). XGBOOST is an efficient and distributed system to improve the gradient tree boosting algorithms. In XGBOOST, the cost function is expanded into two order Taylor's expansion, while the L1 and L2 regularizations are introduced. The regularization of the leaf nodes and column subsampling are used to balance the decline of the cost function and the model complexity in order to avoid overfitting. The step shrinkage multiplied to the leaf weights after each iteration can reduce the effects of trees to expand learning space. It retrofits gradient tree boosting algorithms for handling sparse data, raises a weighted quantile sketch for approximate optimization, and designs a column block structure for parallelization.

Machine Learning hyperparameters grid search

The hyperparameter grid for the different machine learning algorithms was defined as follows:

SVM	› C (cost of constraints violation) = Sequence from 2^{-5} to 2^{15}. › sigma (sigma inverse kernel width for the Radial Basis kernel function "rbfdot") = Sequence from 2^{-15} to 2^{3}.
GBM	› interaction.depth (maximum depth of each tree) = 1. › n.trees (total number of trees to fit) = Sequence from 100 to 3000 by 100. › shrinkage (a shrinkage parameter applied to each tree in the expansion. Also known as the learning rate or step-size reduction) = 0.01.

	› n.minobsinnode (minimum number of observations in the terminal nodes of the trees) = 10.
kNN	› k (number of neighbors) = Sequence from 1 to 50.
cPLS	› ncomp (number of components to be used in the modeling) = Sequence from 1 to 50.
RF	› ntree (number of trees to grow) = Sequence from 50 to 3500 by 50. › mtry (number of variables randomly sampled as candidates at each split) = 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47 or 616.
DL	› epochs = Sequence from 1 to 201 by 20.
XGBOOST	› max_depth (Maximum depth of a tree) = 2, 4 or 6. › nrounds (Maximum number of boosting iterations) = Sequence from 50 to 1000 by 50. › eta (control the learning rate) = 0.003, 0.01 or 0.3. › gamma (minimum loss reduction required to make a further partition on a leaf node of the tree) = 0 or 1. › colsample_bytree (subsample ratio of columns when constructing each tree) = 0.6, 0.8 or 1. › subsample (subsample ratio of the training instance) = 0.5, 0.75 or 1. › min_child_weight (minimum sum of instance weight needed) = 1 or 2.

Variable importance measures

The variable importance (VI) was measured by different techniques depending on the machine learning algorithm used:

› GBM. The relative influence of each variable was assessed using the R package gbm (8) which implements the method described by Friedman *et al* (2).

› XGBOOST. The VI measure represents the fractional contribution of each feature to the model based on the total gain of this feature's splits (6).

› RF. Two VI measures were used: mean decrease in accuracy and mean decrease in node impurity (9).

› cPLS. The VI measure used for this algorithm is based on the weighted sums of the absolute regression coefficients. The weights are a function of the reduction of the sums of squares across the number of PLS components and are computed separately for each outcome. Therefore, the contribution of the coefficients are weighted proportionally to the reduction in the sums of squares (10).

› For the rest of the algorithms, a ROC curve analysis was conducted on each predictor (10). The trapezoidal rule was used to compute the AUROC, and this area was taken as the measure of VI.

REFERENCES

1. James G, Witten D, Hastie T, Tibshirani R. An introduction to statistical learning with applications in R. Springer-Verlag New York; 2013. 426 p.
2. Friedman JH. Greedy function approximation: A gradient boosting machine. Ann Stat. 2001;29(5):1189–232.
3. Barker M, Rayens W. Partial least squares for discrimination. J Chemom. 2003;17(3):166–73.
4. Breiman L. Random Forests. Mach Learn. 2001;45(1):5–32.
5. Ma J, Sheridan RP, Liaw A, Dahl GE, Svetnik V. Deep Neural Nets as a Method for Quantitative Structure–Activity Relationships. J Chem Inf Model. 2015;55(2):263–74.
6. Chen T, He T, Benesty M, Khotilovich V, Tang Y, Cho H, et al. xgboost: Extreme Gradient Boosting. R package version 0.90.0.2. 2019. Available from: https://cran.r-project.org/package=xgboost
7. Chen T, Guestrin C. XGBoost: A Scalable Tree Boosting System. Proc 22nd ACM SIGKDD Int Conf Knowl Discov Data Min - KDD'16. 2016;785–94.
8. Greenwell B, Boehmke B, Cunningham J, GBM D. gbm: Generalized Boosted Regression Models. R pacakge version 2.1.5. 2019.
9. Liaw A, Wiener M. Classification and Regression by randomForest. R news. 2002;3:18–22.
10. Kuhn M, Wing J, Weston S, Williams A, Keefer C, Engelhardt A, et al. caret: Classification and Regression Training. R package version 6.0-84. 2018. p. 216.