Machine Learning for Drug Design

Ying Liu
Division of Computer Science, Mathematics and Science
St. John's University
Queens, NY 11349
Email: liuy1 [AT] stjohns.edu

***Abstract*—A common step in drug design is the formation of a quantitative structure-activity relationship (QSAR) to model an exploratory series of compounds. A QSAR generalizes how the structure of a compound relates to its biological activity. There is growing interest in the application of machine learning techniques in QSAR modeling research. However, no single technique can claim to be uniformly superior to any other. This study introduced the ensemble machine learning, a set of classifiers whose individual decisions are combined in some way (typically by weighted or unweighted voting) to improve the performance of the overall system. A comparative study was carried out of two popular ensemble learning algorithms, Bagging and AdaBoost, for QSAR modeling. Two test case problems were studied: the inhibition of Escherichia coli dihydrofolate reductase (DHFR) by pyrimidines, and the inhibition of rat/mouse tumor DHFR by triazines. It was observed that the ensemble learning algorithms, Bagging and AdaBoost, can significantly improve the performance of Decision Tree C4.5 and 1-R ($p<0.05$), while Naïve Bayesian and 1-Nearest Neighbor did not benefit from ensemble learning. Furthermore, in general, AdaBoost outperformed Bagging on the tested data sets.**

Keywords-machine learning, drug design, QSAR, enselmble learning

I. INTRODUCTION

Quantitative structure-activity relationship (QSAR) analysis represents an essential part of the drug discovery process to reduce the search for new drugs [1]. QSAR is based on the assumption that there exists a relationship between the structural or molecular features of a compound and its biological activity (such as chemical activity, aqueous solubility, blood-brain barrier penetration, oral absorption or toxicity). The aim of QSAR analysis is to discover these relationships in order to predict the activity of new molecules based on their physiochemical descriptors [2, 3, 4].

QSAR analysis is becoming increasingly important in automated pharmaceutical production processes. It also presents an extremely challenging problem to the field of Intelligent Systems and one that, if solved successfully, has the potential to provide significant economic benefit. New compounds emerging from the production lines must be screened for their potential use (measured by chemical or biological activity in some assay) in future products. The capacity of the production lines is increasing through developments in robot technology and pharmaceutical methods. QSAR analysis forms an essential part of the overall screening process, in which new compounds are tested against structural models to determine their potential activity or otherwise [1].

In recent years, artificial intelligence techniques have been applied to model QSAR's, such as neural networks [5-9], genetic algorithms [10], decision trees [11], inductive logic programming [11, 12], and support vector machine [1]. Machine learning techniques have, in general, offered greater accuracy than have their statistical forebears, but not without accompanying problems for the QSAR analysts to consider. Neural networks, for example, offer high accuracy in most cases but can suffer from overfitting the training data [13]. Other problems with the use of neural networks concern the reproducibility of results, owing largely to set-up and stopping criteria, and lack of information regarding the classification produced [13]. Genetic algorithms may also suffer from their stochastic nature, in that results may be hard to reproduce and the resulting classification may not be optimal [14]. Decision trees offer a large amount of information regarding their decisions, in the form of predictive rules, but occasionally struggle to provide the accuracy supplied by more powerful, but less informative, techniques. Owing to the reasons outlined above, there is a continuing need for the application of more accurate and informative classification techniques to QSAR analysis [1].

An ensemble of classifiers is a set of classifiers whose individual decisions are combined in some way (typically by weighted or unweighted voting) to improve the performance of the overall system [15]. Other terminologies found in the literature to denote similar meanings are: multiple classifiers, multi-strategy learning, committee, classifier fusion, combination, aggregation, and integration [16]. The intuitive concept of ensemble learning is that no single classifier can claim to be uniformly superior to any other, and that the integration of several single classifiers will enhance the performance of the final classifier (e.g. accuracy, reliability). Hence, ensemble classifiers are often much more accurate than the individual classifiers that make them up. The easiest approach to generate diverse classifiers is to manipulate the training data and run a base learner on the manipulated training data multiple times. Ensemble learning methods have been shown to be very successful in improving the accuracy of

certain classifiers for artificial and real-world datasets [15-21]. Using decision tree as the base learner, Tan and Gilbert (2003) [16] applied ensemble machine learning to gene expression data for cancer classification. The results showed that ensemble learning performed better than single decision tree. Similar observation was reported by Dietterich (2000b) [21] when 33 different datasets were studied and by Bauer and Kohavi (1999) [20] when 13 datasets were employed.

The focus of this study is to investigate the performance of ensemble machine learning in QSAR modeling. Four base learners, decision tree C4.5, Naïve Bayesian (NB), 1-Nearest Neighbor (1NN), and 1-Rule (1R) [22] were used to construct ensembles.

II. MATERIALS AND METHODS

A. *Notations*

A labeled training example is a pair $<x, y>$ where x is an element from space X and y is an element from a discrete space Y. Let x represent an attribute vector with n attributes and y the class label associated with x for a given example, a classifier (or a hypothesis) is a mapping from X to Y.

B. *Ensemble Machine Learning*

In this study, two of the most popular techniques for constructing ensembles, Bagging and AdaBoost, were investigated. These two techniques manipulate the training examples to generate multiple classifiers. The learning algorithm takes the base learner and a training set as input and runs the base learner multiple times by changing the distribution of the training set instances. The generated classifiers are then combined to create a final classifier that is used to classify the test set [15].

- **Bagging** (**b**ootstrap **agg**regat**ing**) was introduced by Breimen (1996) [23] and it aims to manipulate the training data by randomly replacing the original T training data by N items. The replacement training sets are known as bootstrap replicates in which some instances may not appear while others appear more than once. Each bootstrap replicate contains, on the average, 63.2% of the original training set. The final classifier $C^*(x)$ is constructed by aggregating $C_i(x)$ where every $C_i(x)$ has an equal vote [15, 16, 20].

- **AdaBoost**: Freund and Schapire (1996) [24] introduced the AdaBoost (**Ada**ptive **Boost**ing) method as an alternative method to influence the training data. Initially, the algorithm assigns every instance x_i with an equal weight. In each iteration i, the learning algorithm tries to minimize the weighted error on the training set and returns a classifier $C_i(x)$. The weighted error of $C_i(x)$ is computed and applied to update the weights on the training instances x_i. The weight of x_i increases according to its influences on the classifier's performance that assigns a high weight for a misclassified x_i and a low weight for a correctly classified x_i. The final classifier $C^*(x)$ is constructed by a weighted vote of the individual $C_i(x)$ according to its accuracy based on the weighted training set [15, 16, 20].

C. *Data Sets*

Two well-studied data sets were used as test cases [1, 11, 12, 25, 26]

- **Pyrimidines data set**. A data set of 74 compounds testing inhibition of *Escherichia coli* DHFR by 2,4-diamino-5-(substituted-benzyl)pyrimidines [12, 25]. Each compound has three positions of possible substitution: the 3-, 4-, and 5-positions of the phenyl ring (six-atom carbon ring) [11]. For each substitution position there are nine descriptors: polarity, size, flexibility, hydrogen-bond donor, hydrogen-bond acceptor, π donor, π acceptor, polarizability and σ effect. Each of the twenty-four non-hydrogen substituents was given an integer value for each of these properties [12]; lack of a substitution is indicated by nine -1's. This gives twenty-seven integer attributes for each compound; in addition, each compound has a real valued activity [25]. Therefore, in this dataset, the number of examples is 74, and the dimensionality is 28. This data set was divided randomly into five equal sized cross-validation sets [11].
- **Triazines data set**. A data set of 186 compounds testing inhibition of mouse/rat tumor DHFR by 4,6-diamino-1,2-dihydro-2-2-dimethyl-1(X-phenyl)-s-triazines (ortho-substituents were not considered) [26]. In the triazine compounds, there are six positions of possible substitution: the 3- and 4-positions of the phenyl ring; if the substituent at the 3-position contained a ring itself, then the 3- and 4-positions of the third ring (the attribute values for these regions were set to null if there was no third ring); if the substituent at the 4-position of the phenyl ring contained a ring itself, then the 3- and 4-positions of the third ring (the attribute values were set to null if there was no third ring) [11]. For each substitution position there are ten descriptors: polarity, size, flexibility, hydrogen-bond donor, hydrogen-bond acceptor, π donor, π acceptor, polarizability, σ effect, and branching. This gives sixty integer attributes for each compound; in addition, each compound has a real valued activity [25]. Therefore, in this dataset, the number of examples is 186, and the dimensionality is 61. The triazine data were randomly divided into six equally sized cross-validation sets [11].

D. *Preformance Evaluations*

The performance of Bagging and Boosting compared with the base learners were measured using several statistics:

- Accuracy: the proportion of correctly classified instances:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

where true positives (TP) denote the correct classifications of positive examples; true negatives (TN) are the correct

classifications of negative examples; false positives (FP) represent the incorrect classification of negative examples into the positive class; and false negatives (FN) are the positive examples incorrectly classified into the negative class.

- Sensitivity: the percent of positive examples which were correctly classified;

$$Sensitivity = \frac{TP}{TP + FN}$$

- Specificity: the percent of negative examples which were correctly classified;

$$Specificity = \frac{TN}{TN + FP}$$

- Positive Predictive Value (PPV): the percentage of the examples predicted to be positive that were correct;

$$PPV = \frac{TP}{TP + FP}$$

- Negative Predictive Value (NPV): the percentage of the examples predicted to be negative that were correct;

$$NPV = \frac{TN}{TN + FN}$$

III. Results

A. Pyrimidine data set

The performance of the ensembles and the base learners on the pyrimidine data set was shown in Figure 1. Bagging and AdaBoost significantly improved the performance of decision tree C4.5 as measure by sensitivity, specificity, and accuracy ($p<0.05$). Furthermore, AdaBoost also improved the PPV of C4.5 ($P<0.05$). 1R also benefited from AdaBoost. When AdaBoost was applied to the pyrimidine data set, the sensitivity, specificity, PPV, and accuracy were significantly improved from 0.76, 0.65, 0.55, and 0.66 to 0.83 ($p<0.05$), 0.84 ($p<0.01$), 0.84 ($p<0.05$), and 0.83 ($p<0.05$), respectively, (Figure 1B). From Figure 1C, it can be observed that AdaBoost NB performed better than single NB. However, the difference was not significant ($p>0.05$). When 1NN was used as the base learner, there was no significant difference between the ensemble results and the 1NN result ($P>0.05$) (Figure 1D).

B. Triazine data set

Similar to the results for the pyrimidine data set, Bagging and AdaBoost improved the performance of decision tree C4.5 and 1R significantly ($p<0.05$) when the algorithms were applied to triazine data set (Figure 2A and Figure 2B). Whereas, the ensembles did not outperform the base learners, NB and 1NN ($p>0.05$) (Figure 2C and Figure 2D).

In general, the results for the pyrimidine data set were better than those for the triazine data set in terms of sensitivity, specificity, PPV, NPV, and accuracy (Figure 1 and Figure 2). Take accuracy as an example, for pyrimidine data set, the results obtained from all the methods had an accuracy over 0.83 except 1R and Bagging 1R, while for triazine data set, the results obtained from all the methods had accuracy less than 0.80 except Bagging C4.5 and AdaBoost C4.5. This is probably due to the simpler structure of the pyrimidine compounds [11].

IV. Discussion

The advent of combinatorial chemistry in the mid-1980s has allowed the automatic synthesis of millions of new molecular compounds. The need for a more refined search methodology than simply producing and testing every single molecular combination possible has meant that statistical approaches and, more recently, intelligent computation have become an integral part of the drug production process. QSAR analysis is one technique used to reduce the search for new drugs. Machine Learning techniques have already started to be successfully applied to the problem of SAR analysis [4]. However, no single technique can claim to be uniformly superior to any other. One of the most active areas of research in machine learning has been to study methods for constructing good ensembles.

A. Ensembles outperform single learning algorithms

Studies have been repeatedly demonstrated significant performance improvements through ensemble methods. There are three fundamental reasons for this:

The first reason is statistical. A learning algorithm can be viewed as searching a space *H* of hypotheses to identify the best hypothesis in the space. The statistical problem arises when the amount of training data available is too small compared to the size of the hypothesis space. Without sufficient data, the learning algorithm can find many different hypotheses in *H* that all give the same accuracy on the training data. By constructing an ensemble out of all of these accurate classifiers, the algorithm can “average” their votes and reduce the risk choosing the wrong classifier.

The second reason is computational. Many learning algorithms work by performing some form of local search that may get stuck in local optima. In cases where there is enough training data (so that the statistical problem is absent), it may still be very difficult computationally for the learning algorithm to find the best hypothesis. An ensemble constructed by running the local search from many different starting points may provide a better approximation to the true unknown function than any of the individual classifiers.

The third reason is representational. In most applications of machine learning, the true classification function cannot be represented by any of the hypotheses in H. By forming weighted sums of hypotheses drawn from *H*, it may be possible to expand the space of representable functions.

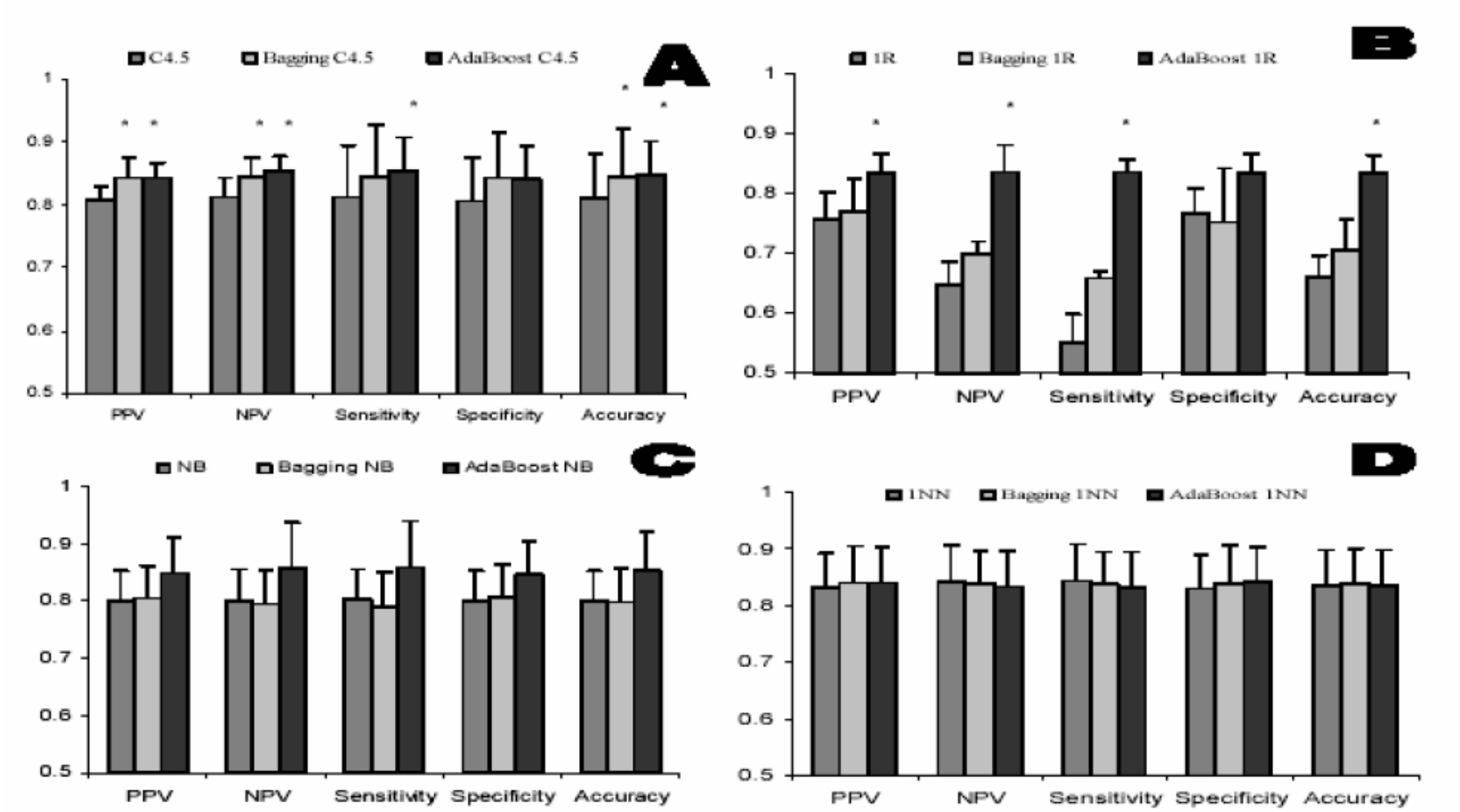

Figure 1. Pyrimidines data set result. The bar chart compared mean accuracy, mean sensitivity, mean specificity, mean positive predictive value (PPV), and mean negative predictive value (NPV) for ensemble learning algorithms, Bagging and AdaBoost, and single learning algorithms, decision tree C4.5 (A), 1-R (B), Naïve Bayesian (C), and 1-Nearest Neighbor (D). Error bars indicated standard errors. The significant difference between the results obtained from the ensembles and those obtained from the single base learners were showed with *.

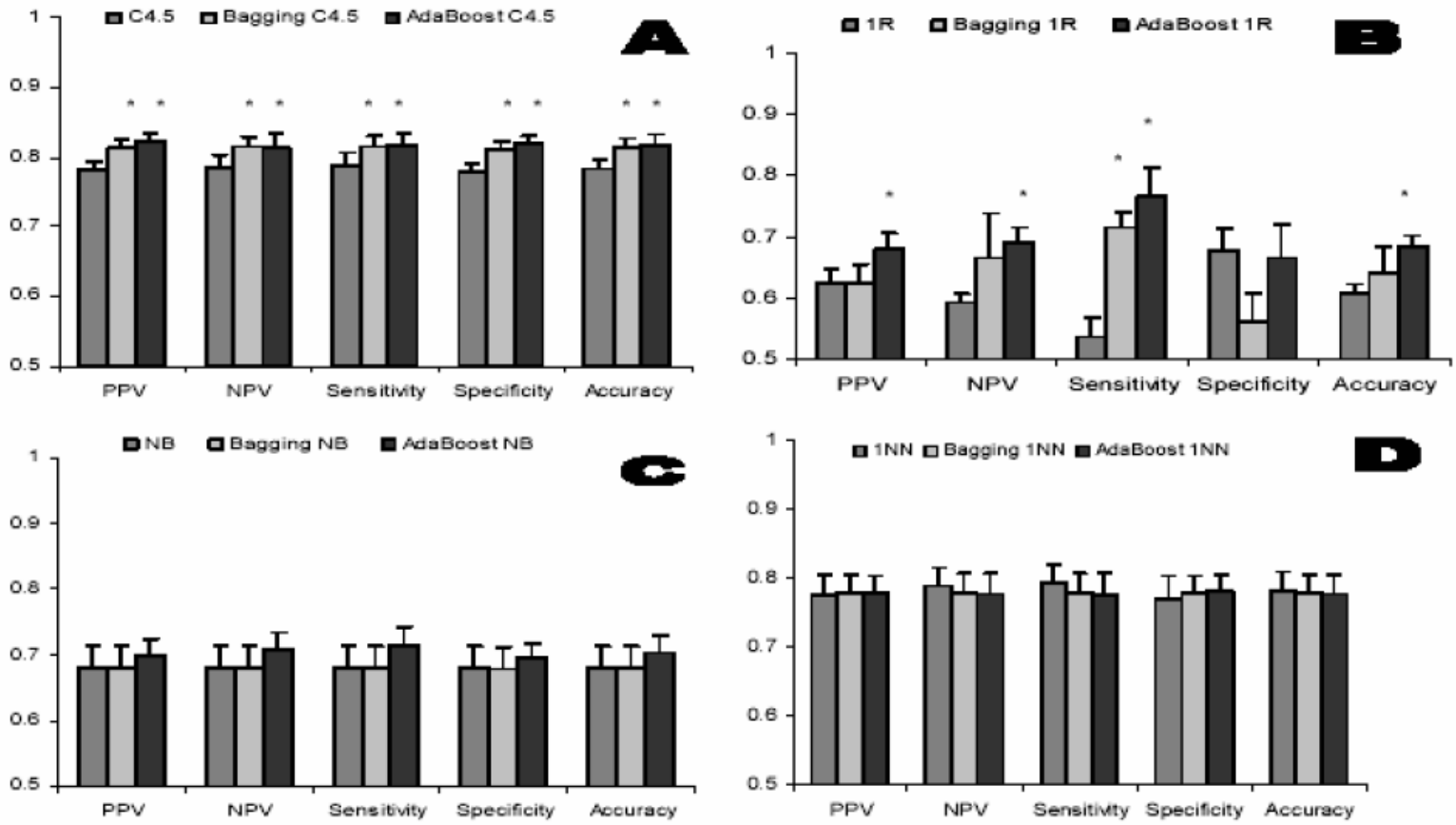

Figure 2. Triazines data set result. The bar chart compared mean accuracy, mean sensitivity, mean specificity, mean positive predictive value (PPV), and mean negative predictive value (NPV) for ensemble learning algorithms, Bagging and AdaBoost, and single learning algorithms, decision tree C4.5 (A), 1-R (B), Naïve Bayesian (C), and 1-Nearest Neighbor (D). Error bars indicated standard errors. The significant difference between the results obtained from the ensembles and those obtained from the single base learners were showed with *.

These three fundamental issues are the three most important ways in which existing learning algorithms fail. Ensemble methods have the promise of reducing (and perhaps even eliminating) these three key shortcomings of standard learning algorithms [15].

B. Base learner effect on the performance of ensemble learning methods

A necessary and sufficient condition for an ensemble learning algorithm to be more accurate than any of its individual members is whether the algorithms are accurate and diverse [15, 21]. Dietterich (2000a) [15] claimed that Bagging and AdaBoost work especially well for unstable learning algorithms – algorithms whose output classifier undergoes major changes in response to small changes in the training data. Decision tree, and rule learning algorithms are unstable. Nearest Neighbor, Naïve Bayesian algorithms are generally very stable [15, 20]. This study confirmed the claim. Bagging and AdaBoost performed well when decision tree C4.5 and 1R were used as the base learners, while they did not improve the performance when 1NN and NB were used as the base learners. Other studies reported similar results [15, 17, 18, 20, 21, 23].

C. AdaBoost outperformed Bagging

In this study, AdaBoost, in general, outperformed Bagging (Figure 1 and Figure 2). AdaBoost, like Bagging, manipulates the training data to generate multiple hypotheses. AdaBoost maintains a set of weights over the training examples. In each iteration, the learning algorithm is invoked to minimize the weighted error on the training examples, and it returns a classifier. Dietterich (2000a) [15] pointed out that, in low-noise cases, AdaBoost gives good performance, because it is able to optimize the ensemble without overfitting. However, in high-noise cases, AdaBoost puts a large amount of weight on the mislabled examples, and this leads it to overfit very badly. Bagging does well in both noisy and noise-free cases.

V. CONCLUSION

Machine learning has increasingly gained attention in drug discovery research. Ensemble machine learning has been an active research topic in machine learning but is still relatively new to the drug discovery research community. Most of the machine learning oriented drug discovery research still largely concentrates on single learning approaches. It is believed that ensemble learning is suitable for drug design applications due to the fact that the classifiers are induced from incomplete and noisy data.

In this study, we have demonstrated the usefulness of employing ensemble methods in QSAR modeling. Ensemble learning methods have been shown to be very successful in improving the accuracy of certain classifiers, i.e., decision tree C4.5 and 1R.

REFERENCES

[1] Burbidge, R., Trotter, M., Buxton, B., 2001. Drug design by machine learning: support vector machines for pharmaceutical data analysis. Computers and Chemistry. 26, 5-14.
[2] Frimurer, T. M., Bywater, R., Narum, L., Lauritsen, L. N., Brunak, S., 2000. Improving the odds in discriminating drug-like from non drug-like compounds. Journal of Chem. Inf. Comput. Sci. 40, 1315-1324.
[3] Wagener, M., van Geerestein, V., 2000. Potential drugs and non-drugs: prediction and identification of important structural features. Journal of Chem. Inf. Comput. Sci. 40, 280-292.
[4] Liu, Y., 2004. A comparative study on feature selection methods for drug discovery. Journal of Chem. Inf. Comput. Sci. 44, 1823-1828.
[5] Anoyama, T., Suzuki, Y., Ichikawa, H., 1990. Neural networks applied to structure-active relationships. Journal of Medicinal Chemistry. 33, 905-908.
[6] Andrea, T. A., Kalayeh, H., 1991. Application of neural networks in quantitative structure-activity relationships of dihydrofolate reductase inhibitors. Journal of Medicinal Chemistry. 34, 2824-2836.
[7] Anoyama, T., Ichikawa, H., 1992. Neural networks as nonlinear structure-activity relationship analysers: useful functions of the partial derivative methods in multilayer neural networks. Journal of Chemical Information and Computer Sciences. 32, 592-500.
[8] So, S.-S., Richards, W. G., 1992. Application of neural networks: Quantitative structure-activity relationships of the derivatives of 2,4-diamino-5-(substituted-benzyl)pyrimidines as DHFR inhibitors. Journal of Medicinal Chemistry. 35, 3201-3207.
[9] Devillers, J. 1999a. Neural Networks and Drug Design. Academic Press.
[10] Devillers, J., 1999b. Genetic Algorithms in Molecular Modeling. Academic Press.
[11] King, R. D., Hirst, J. D., Sternberg, M. J. E., 1995. Comparison of artificial intellogence methods for modeling pharmaceutical QSARs. Applied Artificial Intelligence. 9, 213-233.
[12] King, R. D., Muggleton, S., Lewis, R. A., Sternberg, M. J. E., 1992. Drug design by machine learning: the use of inductive logic programming to model the structure-activity relationships of trimethoprim analogues binding to dihydrogolate reductase. Proceeding of National Academy of Science USA. 89, 11322-11326.
[13] Manallack, D. T., Livingstone, D. J., 1999. Neural networks in drug discovery: have they lived up to their promise? Eur. J. Med. Chem. 34, 95-208
[14] Goldberg, D., 1989. Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley.
[15] Dietterich, T. G., 2000a. Ensemble methods in machine learning. In J. Kittler and F. Roli, editors, Multiple Classifier Systems. First International Workshop, MCS 2000, Cagliari, Italy, volume 1857 of Lecture Notes in Computer Science. pages 1–15. Springer-Verlag.
[16] Tan, A. C., Gilbert, D., 2003. Ensemble machine learning on gene expression data for cancer classification. Applied Bioinformatics. 2, S75-S83.
[17] Drucker, H., Cortes, C., 1996. Boosting decision trees. Advances in Neural Information Processing Systems. 8, 479-485.

[18] Quinlan, J. R., 1996. Bagging, boosting, and c4.5. Proceeding of the 13th International Conference on Artificial Intelligence. p725-730.
[19] Elkan, C., 1997. Boosting and naïve Bayesian learning. Technical report, Department of Computer Science and Engineering, University of California, San Diego.
[20] Bauer, E.; Kohavi, R., 1999. An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. Machine Learning. 36, 105-139.
[21] Dietterich, T. G., 2000b. An experimental comparison of three methods for constructing ensembles of decision tress: Bagging, boosting and randomization. Machine Learning. 40, 139–157
[22] Witten, I. H., Frank, E., 2000. Data mining: Practical Machine Learning Tools and Techniques with Java Implementations. Morgan Kaufmann Publishers. San Francisco, USA.
[23] Breiman, L., 1996. Bagging predictors. Machine Learning. 24, 123-140.
[24] Freund, Y., Schapire, R. E., 1996. Experiments with a new boosting algorithm. Proceeding of 13th International Conference on Machine Learning. p148-156
[25] Hirst, J. D., King, R. D., Sternberg, M. J. E., 1994a. Quantitative structure-activity relationships: Neural networks and inductive logic programming compared against statistical methoids: I, the inhibition of dihydrofolate reductase by pyrimidines. Journal of Computer-Aided Molecular Design. 8, 405-420.
[26] Hirst, J. D., King, R. D., Sternberg, M. J. E., 1994b. Quantitative structure-activity relationships: Neural networks and inductive logic programming compared against statistical methoids: II, the inhibition of dihydrofolate reductase by triazines. Journal of Computer-Aided Molecular Design. 8, 421-432.

.