An Integrated Approach to Feature Invention and Model Construction for Drug Activity Prediction

Jesse Davis JDAVIS@CS.WISC.EDU
Department of Computer Science, University of Wisconsin-Madison, USA
Vítor Santos Costa VSC@NCC.UP.PT
LIACC and DCC/FCUP, Universidade do Porto, Portugal
Soumya Ray SRAY@EECS.OREGONSTATE.EDU
School of Electrical Engineering and Computer Science, Oregon State University, USA
David Page PAGE@BIOSTAT.WISC.EDU
Department of Biostatistics and Medical Informatics, University of Wisconsin-Madison, USA

Abstract

We present a new machine learning approach for 3D-QSAR, the task of predicting binding affinities of molecules to target proteins based on 3D structure. Our approach predicts binding affinity by using regression on substructures discovered by relational learning. We make two contributions to the state-of-the-art. First, we use multiple-instance (MI) regression, which represents a molecule as a set of 3D conformations, to model activity. Second, the relational learning component employs the "Score As You Use" (SAYU) method to select substructures for their ability to improve the regression model. This is the first application of SAYU to multiple-instance, real-valued prediction. We evaluate our approach on three tasks and demonstrate that (i) SAYU outperforms standard coverage measures when selecting features for regression, (ii) the MI representation improves accuracy over standard single feature-vector encodings and (iii) combining SAYU with MI regression is more accurate for 3D-QSAR than either approach by itself.

1. Introduction

Recent studies in relational learning have shown the effectiveness of combining learned relational rules using a statistical classifier. The most successful of these approaches actually score candidate relational rules, during relational learning, by their ability to improve the statistical classifier. In the first such studies—with systems nFOIL (Landwehr et al., 2005) and "Score As You Use" (SAYU) (Davis et al., 2005)—the classifier is a form of Bayesian network, the rules act as binary features in the network, and the class value acts as a binary class feature. A later system, kFOIL (Landwehr et al., 2006), used a kernel as the model, which allows it to handle both classification and regression.

The present paper extends these approaches to multiple-instance, real-valued prediction. While such prediction is useful for many applications, our motivating application is predicting Three-dimensional Quantitative Structure-Activity Relationships (abbreviated as 3D-QSARs). This task comes from a well-known family of tasks in the pharmaceutical industry and in research into drug design. Each 3D-QSAR task is defined by a target protein, typically whose 3D structure is not known. Given the structures of a set of molecules and their known binding affinities to the target, the task is to construct a model that accurately predicts the real-valued binding affinities of new small molecules to the target, based on their three-dimensional structures. Our paper makes two contributions. First, it demonstrates the feasibility and utility of extending approaches such as SAYU to multiple-instance, real-valued prediction, using a significant real-world application. Second, the paper shows empirically that using multiple-instance (MI) regression in this context carries significant benefit over using ordinary regression.

In prior work, relational learning combined with linear regression has been applied with some success to 3D-QSAR (Marchand-Geneste et al., 2002). This approach represents molecules using clauses in first-order logic. From this representation, the algorithm learns rules that represent potential *pharmacophores*—3D substructures of molecules re-

Appearing in *Proceedings of the 24^{th} International Conference on Machine Learning*, Corvallis, OR, 2007.

sponsible for binding. This approach then treats each rule as an attribute and constructs one feature vector for each molecule. Finally, it constructs a regression model for real-valued activity prediction from the resulting feature vectors. This approach has two shortcomings. First, even though the goal is to optimize the accuracy of the real-valued prediction, the relational learning procedure is not guided by this goal but rather by a different scoring function that is usually based on the coverage of the rules on the training set. Second, by operating on one feature vector per molecule, regression ignores the inherent multiple-instance nature of 3D molecular data (Dietterich et al., 1997). The MI nature arises from the fact that each molecule may have multiple low-energy 3D shapes, or *conformers*, and any one (or several) of these conformers may be the one that binds to the target. Information about the individual conformers is lost when we construct one feature vector per molecule.

We propose a framework that addresses the above issues. First, during the search for potential pharmacophores, we score each potential pharmacophore by how much it improves the generalization ability of the regression model. We accomplish this by adapting the SAYU approach (Davis et al., 2005) to this regression task. For each potential pharmacophore, or rule, that must be scored, we re-compute the regression model and check whether it generalizes better than the model that does not use the candidate rule. Thus, a pharmacophore is included in the regression model only if it helps to predict the observed activities. Second, we adapt multiple-instance regression (Ray & Page, 2001) to the task of 3D-QSAR. MI regression replaces standard regression when predicting activity. Here, when we evaluate a set of rules, we construct one feature vector per conformation rather than one feature vector per molecule. MI regression is able to operate on data of this form. We evaluate our proposed approach on three tasks, dopamine (D2 receptor) agonists, thermolysin inhibitors and thrombin inhibitors, and demonstrate that our approach results in more accurate predictions than state-of-the-art methods that combine relational learning with regression for 3D-QSAR.

2. Background and Related Work

Drugs are small molecules that affect disease by binding to a target protein in the human body; the target may be a human protein or a protein belonging to some pathogen that has entered the body. A key obstacle in the drug design process is that the structure of a target protein cannot often be determined. In this scenario, researchers may use "high-throughput screening" to test a large number of small molecules to find some that bind to the target. The molecules that bind usually cannot be used as drugs for various reasons, typically related to ADMET (absorption, distribution, metabolism, elimination and toxicity). Nevertheless, some of these "hits" may be used as a "lead" in the search for an appropriate drug; this lead then needs to be "optimized," or modified in order to have an appropriate binding affinity and the other properties necessary in a drug, such as low side effects.

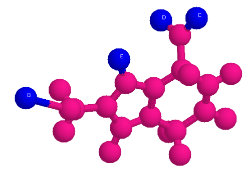

Figure 1. ACE inhibitor with highlighted 4-point pharmacophore.

To guide the lead optimization process, researchers try to find similarities between the most active molecules that are ideally not shared by any of the less active molecules. We briefly describe the types of similarities that are useful in predicting binding affinity. A small molecule binds to a protein primarily based on electrostatic and hydrophobic interactions. The most common electrostatic interaction is the hydrogen bond, where an atom carrying a slight negative charge, such as an oxygen (a "hydrogen acceptor"), on one molecule is attracted to a hydrogen atom carrying a slight positive charge (a "hydrogen donor") on the other molecule. Hydrophobic interactions typically occur when hydrophobes from the two molecules shield each other from the surrounding aqueous environment. Because both electrostatic and hydrophobic interactions are weaker than the ordinary covalent bonds formed within a molecule, several such interactions—typically three to eight—are required in order for a small molecule to bind to a protein. Therefore, to bind to a given target protein at a particular site, a small molecule needs the right combination of charged atoms and/or hydrophobic groups *at the right locations*. In other words, the binding sites on the small molecule and protein need to be complementary, much as a key is to a lock—a common analogy in drug design. Given a set of active molecules, a computational chemist may search for conformers of the active molecules that share some three-dimensional arrangement of charged atoms, such as potential hydrogen donors and acceptors, and hydrophobic groups, such as six-membered carbon rings. This three-dimensional substructure is sometimes called a *pharmacophore*. Figure 1 shows an example molecule with a highlighted pharmacophore that allows it to inhibit Angiotensin-Converting Enzyme (ACE).

3D-QSAR approaches directly address the multiple 3D conformers of molecules. These approaches include both special-purpose algorithms for 3D-QSAR and mechanisms for applying machine learning algorithms to the multiple 3D conformers of molecules. CoMFA and related

Table 1. An example of a 4-point pharmacophore learned by ALEPH for the domain of thermolysin inhibitors. The left column shows the first-order logical clause in Prolog notation, while the right column shows the semantics of each literal.

`active(M):-`	Molecule M is active if
`conf(M, C),`	M has a conformation C
`hacc(M, C, P1),`	C has a hydrogen acceptor at location $P1$
`hacc(M, C, P2),`	C has a hydrogen acceptor at location $P2$
`hacc(M, C, P3),`	C has a hydrogen acceptor at location $P3$
`pos_charge(M, C, P4),`	C has a positively charged group at location $P4$
`dist(M, C, P1, P2, 4.60, 1.0),`	the distance between $P1$ and $P2$ is $4.60 \pm 1.0\mathring{A}$
`dist(M, C, P1, P3, 7.75, 1.0),`	the distance between $P1$ and $P3$ is $7.75 \pm 1.0\mathring{A}$
`dist(M, C, P2, P3, 8.77, 1.0),`	the distance between $P2$ and $P3$ is $8.77 \pm 1.0\mathring{A}$
`dist(M, C, P1, P4, 6.85, 1.0),`	the distance between $P1$ and $P4$ is $6.85 \pm 1.0\mathring{A}$
`dist(M, C, P2, P4, 7.56, 1.0),`	the distance between $P2$ and $P4$ is $7.56 \pm 1.0\mathring{A}$
`dist(M, C, P3, P4, 1.24, 1.0).`	the distance between $P3$ and $P4$ is $1.24 \pm 1.0\mathring{A}$

approaches rely on careful feature construction based on structural properties at grid points defined on the molecule's surface (for example, see Cramer *et al.* (1988)). DISCO (Martin et al., 1993) uses a clique detection algorithm (Brint & Willett, 1987) to predict pharmacophores from the conformers of active molecules. The COMPASS algorithm (Jain et al., 1994a; Jain et al., 1994b) selects and aligns conformers—one per active molecule—and generates a feature vector for each molecule, where the features are the lengths of rays passing through the molecule at specified orientations. COMPASS then uses a neural network to learn either a classifier or real-valued predictor of affinities; once the model has been learned, COMPASS tries to improve the fit by revisiting the selection and alignment of conformers, and iterates until convergence.

Another approach to predicting binding affinities is based on relational learning (Finn et al., 1998; Marchand-Geneste et al., 2002), in particular inductive logic programming (ILP). This ILP-based framework starts with a first-order logical description of each molecule. This description details the locations of each atom and bond in the molecule. Additionally, the background knowledge contains relational descriptions of common groups of atoms. For example, the background knowledge can specify that a methyl group consists of a carbon atom bound to three hydrogen atoms with single bonds. A *k-point pharmacophore* in this representation is a first-order clause that has k literals, each describing a distinct chemical group (such as methyl), and $\binom{k}{2}$ "distance" literals. Each distance literal stores the Euclidean distance between two chemical groups. Since the distances in any two given molecules are unlikely to be exactly the same, the literal includes a tolerance that specifies how much each distance is allowed to vary. Given this representation, the approach uses an ILP system to hypothesize pharmacophores that cause the desired interaction between known active molecules and the target. The ILP system searches over the space of clauses (pharmacophores) using an objective function such as the following: any k-point pharmacophore that appears significantly more often in active molecules than in inactive ones is hypothesized to be an interaction-causing pharmacophore. Table 1 shows an example pharmacophore learned by an ILP system, ALEPH[1], for the domain of thermolysin inhibitors.

In order to predict real-valued activities, the ILP-based approach treats learned clauses as binary-valued features and generates a binary (0/1) value depending on whether that molecule satisfies the given clause (i.e., has the specified pharmacophore in *any* conformation). Of course, using this representation, the inactive (or "less active") molecules will have features that are mostly zero, which will likely lead to poor activity estimates. Thus, the ILP-based approach also learns a set of features that are more frequent in the inactive molecules than in the active molecules, and generate the corresponding feature vectors. This procedure generates a single feature vector for each molecule. This representation can then be used to learn a regression model using standard linear regression (Marchand-Geneste et al., 2002), which can predict activity levels for novel molecules.

3. The MIR-SAYU Algorithm

Our algorithm follows the ILP-based approach just described, but with two significant changes. First, it uses the "Score As You Use" method to learn rules directly judged as helpful to regression. Second, it constructs one feature vector per conformer rather than one per molecule, and it then employs multiple-instance regression to predict binding affinity. In the following sections, we describe each component of this approach in detail.

3.1. Scoring Candidate Rules with SAYU

In relational approaches to 3D-QSAR, as described above, an ILP system generates rules describing pharmacophores. In prior work (Finn et al., 1998; Marchand-Geneste et al., 2002), this system runs to completion, and a subset of the rules found are used to build the model. This approach relies on the ILP system's score metric to evaluate rule quality. The most common metric is *coverage*, which is defined

[1] ALEPH is an ILP system written by Ashwin Srinivasan.

as the difference between the number of active and inactive molecules that satisfy a rule. The final model is built from the rules which have the highest coverages. This approach has several drawbacks. First, running to completion may take a long time. Second, the rules may not be independent, leading to a model with dependent attributes. Third, choosing how many rules to include in the final model is a difficult tradeoff between completeness and overfitting. Finally, the best rules according to coverage may not give us the most accurate activity model.

Many of these drawbacks can be overcome by interleaving the rule learning and model building processes. In our work, we accomplish this interleaving by extending the SAYU approach (Davis et al., 2005) to the MI regression setting. In the SAYU approach, we start from an empty model (or a prior model). Next, an ILP system generates rules, each of which represents a new feature to be added to the current model. We then evaluate the generalization ability of the model extended with the new feature. We retain the new model if the addition of the new feature improves the model's generalization ability; otherwise we remain with the original model. This results in a tight coupling between feature construction and model building.

To apply SAYU to our task, we need an ILP system to propose rules. In our work, we use ALEPH, which implements the Progol algorithm (Muggleton, 1995) to learn rules. This algorithm induces rules in two steps. Initially, it selects a positive instance to serve as the "seed" example. It then identifies all the facts known to be true about the seed example. The combination of these facts forms the example's most specific or saturated clause. The key insight of the Progol algorithm is that some of these facts explain this example's classification. Thus, generalizations of those facts could apply to other examples. ALEPH therefore performs a general to specific search over the set of rules that generalize a seed example's saturated clause. Thus, in our application, ALEPH picks an "active" molecule, and generates potential k-point pharmacophores from it. It continues until either finding a potential pharmacophore that has high coverage or the search space is exhausted. In the latter case, the search restarts with a different seed.

SAYU modifies the standard ALEPH search as follows. In contrast to ALEPH, SAYU allows any example, positive or negative, to be selected as a seed, because it is possible for the generalization of any example to improve the final regression model. Instead of using coverage, ALEPH passes each clause it constructs to SAYU, which converts the clause to a binary feature and adds it to the current training set. Next, SAYU learns a model incorporating the new feature, and evaluates the model (described below). If the model does not improve, the rule is not accepted, and control returns to ALEPH to construct the next clause. If a rule is accepted, or the search space is exhausted, SAYU randomly selects a new seed and re-initializes ALEPH's search. Thus, we are not searching for the best rule, but the first rule that improves the model. However, SAYU allows the same seed to be selected multiple times during the search. Since the search space is extremely large, it is impractical to search it exhaustively. Further, this may lead to overfitting. Therefore, as in prior work (Davis et al., 2005), we terminate the search after a certain amount of time.

In order to decide whether to retain a candidate feature f, we need to estimate the generalization ability of the model with and without the new feature. In our work, we do this by estimating the test-set r^2 of each model, defined as:

$$\text{Test-set } r^2 = 1 - \frac{\sum_i (Y_i - p_i)^2}{\sum_i (Y_i - a_i)^2}, \tag{1}$$

where i ranges over test examples, Y_i denotes the true response of the i^{th} test example, p_i denotes the predicted response of the i^{th} test example using our model, and a_i denotes the average response on the training set. Thus, r^2 measures the improvement in squared error obtained by using our model over a baseline constant prediction. Observe that if $p_i = a_i$, $r^2 = 0$, and if $p_i = Y_i$, $r^2 = 1$. Thus, a higher test-set r^2 indicates a model with better generalization ability. Note though that unlike ordinary r^2, it is possible for test-set r^2 to be negative, since predictions are made on novel data points. To estimate test-set r^2 for our models, we use *internal n-fold cross validation* on our training set. In turn, we hold out one fold and learn a model using the remaining folds. We use the model to make predictions on the held-out data. At the end of this procedure, we have a set of predictions for each held-out fold. We then pool these predictions across all folds and calculate the test-set r^2 metric for the model containing f over the full set of predictions. To decide whether to retain the candidate feature, we stipulate that the test-set r^2 of the model with f must improve over the model without f by a certain fraction, p. We call p the improvement threshold.[2] While such cross-validation is computationally expensive, we have observed that it significantly improves the quality of the features added to our models and reduces overfitting to the training set. Further, since we impose an external time constraint on SAYU as described in the previous paragraph, this procedure does not slow down our empirical evaluation. After a set of features have been selected using the cross-validation procedure, we learn the final model, which incorporates all selected features, using the entire training set. We use this model to make predictions on unseen examples. This procedure prevents features that do not help

[2] A more principled solution might be to use a statistical hypothesis test between estimates of the test-set r^2 measures of the two models. We have tried this; however, since we generally have very small samples in our experiments, we did not obtain consistent results with this approach.

predict activity from being added to the regression model. However, note that it may add features which help explain low activity. Such features will be associated with negative coefficients in the regression model.

3.2. Predicting Activity with MI Regression

In this section, we present our multiple-instance regression model that we use to predict the activity of molecules. These are the models that the SAYU procedure constructs when evaluating candidate features.

The relational learning procedure described in Section 2 results in a single feature vector describing each molecule. In prior work (Marchand-Geneste et al., 2002), these features have been used as inputs to a linear regression procedure to predict activity. Linear regression on these features will be effective in predicting activity if the following assumption holds: *the activity of a molecule is a linear function of the pharmacophores it has in at least one of its conformations*. This assumption is somewhat unsatisfactory, as we treat molecules where all pharmacophores match the same conformation(s) and molecules where each pharmacophores matches a different conformation in exactly the same way. Chemically, activity is likely to be a function of specific conformation(s) of the molecule, and this information has been lost. To capture this knowledge, we use a *multiple-instance* representation (Dietterich et al., 1997). In MI learning, examples are represented using *multisets* of feature vectors instead of single feature vectors. In MI terminology, each example is a *bag* of *instances*. Each bag is associated with a class label in a classification setting or a real-valued response in a regression setting. Given this representation, MI algorithms can learn models that predict the class label or response of novel bags.

To generate an MI representation for the drug activity prediction problem, we apply the proposed clauses (pharmacophores) *to each conformation* of each molecule separately. In this case, a 0/1 value represents whether a *specific conformation* has the given pharmacophore (clause). This creates an MI representation, where each molecule is represented by a bag of feature vectors, one per conformation, and the bag is labeled with the activity of the molecule. Given this representation, we use a multiple-instance regression algorithm to learn linear models. The task under consideration is defined as follows. We are given a set of n bags. The i^{th} bag consists of m_i instances and a real-valued response y_i. Instance j of bag i is described by a real-valued attribute vector $\vec{X}_{ij}$ of dimension d. In the drug design example, each bag is a molecule, and each instance a conformation of the molecule represented by a feature vector. An iterative algorithm was presented in prior work (Ray & Page, 2001) to learn a linear model $\widehat{\mathbf{b}}$ under the assumption that there is some *primary* instance in each bag which is responsible for the real-valued label:

$$\widehat{\mathbf{b}} = \arg\min_{\mathbf{b}} \sum_{i=1}^{n} (y_i - \vec{X}_{ip} \cdot \mathbf{b})^2, \tag{2}$$

where $\vec{X}_{ip}$ is the feature vector describing the primary instance of bag i, and y_i is the response of bag i. The algorithm presented in prior work iterates between estimating the primary instance in each bag and solving the resulting linear regression problem until convergence. Recent work (Srinivasan et al., 2006) has used this approach to model drug activity, but has had limited empirical success.

Our approach extends this formulation to be more specific to activity prediction in the following way. Instead of assuming that a single, primary conformer is responsible for the activity of the molecule, we assume that the molecule's activity is a *nonlinear weighted average* of the activities of its conformers. Biologically, each conformer can contribute to activity, but the contribution of a conformer dies off exponentially with goodness of fit between conformer and target. Thus, typically, the activity of a molecule will be dominated by its most active conformers. To model this scenario, we use a *softmax* function, denoted by S below:

$$S_\alpha(x_1, \ldots, x_n) = \frac{\sum_{1 \le i \le n} x_i e^{\alpha x_i}}{\sum_{1 \le i \le n} e^{\alpha x_i}}. \tag{3}$$

The input to this function is the predicted activities of the conformation of any molecule. The output is a weighted average of the predicted activities, with the average being dominated by the most active conformation(s). As the parameter α is increased, the output approximates the highest activity more closely. The softmax function has been used in prior work on MI classification as well (Maron, 1998). Thus, it is a suitable choice both from the biological and the MI perspectives. Further, note that the function is differentiable with respect to its inputs. This lets us use a gradient-based optimization procedure to solve for the best linear model $\widehat{\mathbf{b}}$ as follows:

$$\begin{aligned} \widehat{\mathbf{b}} = {} & \arg\min_{\mathbf{b}} \textstyle\sum_{i=1}^{n} (y_i - S_\alpha(\vec{X}_{i1} \cdot \mathbf{b}, \cdots, \vec{X}_{im_i} \cdot \mathbf{b}))^2 \\ & +\lambda ||\mathbf{b}||^2. \end{aligned} \tag{4}$$

Here, y_i represents the activity of the i^{th} molecule, and the predicted activity of conformation j of molecule i is defined by the linear function $\vec{X}_{ij} \cdot \mathbf{b}$. Thus, the first part of this objective specifies that we are searching for the linear model such that the total error between the weighted averages of the predicted conformation activities and the known molecular activities is minimized. The second part of the objective is a regularization factor proportional to $||\mathbf{b}||^2$. Incorporating such a factor is known to reduce overfitting to the training data, and thus improve generalization ability (Vapnik, 1999). We expect our approach to be more accurate than standard regression if (i) the activity of any *conformation* is a linear function of the pharmacophores it has, and (ii) the activity of any molecule is (approximately) an exponentially weighted average of the activities of its individual conformers.

The objective function in Equation 4 is nonlinear and non-convex; hence, standard gradient-based optimization algorithms are susceptible to local minima. To reduce the possibility of being misled by local minima, we employ the technique of random restarts: when learning a model, the optimization algorithm is restarted several times from different, randomly chosen starting points and allowed to run to completion. The final solution returned is the one resulting in the lowest objective function value.

We call our approach, which combines the SAYU procedure with MI regression models, Multiple Instance Regression-SAYU, abbreviated as MIR-SAYU.

4. Empirical Evaluation

In this section, we evaluate our approach on three real-world activity prediction tasks: thermolysin inhibitors, dopamine agonists and thrombin inhibitors. We first describe the domains and characteristics of the datasets we use and then present and discuss our experimental results.

Tasks. The thermolysin inhibitors dataset we use is described in previous work (Marchand-Geneste et al., 2002). Thermolysin belongs to the family of metalloproteases and plays roles in physiological processes such as digestion and blood pressure regulation. The molecules in our dataset are known inhibitors of thermolysin. Activity for these molecules is measured in $pK_i = -\log K_i$, where K_i is a dissociation constant measuring the ratio of the concentrations of bound product to unbound constituents. A higher value indicates a stronger affinity for binding. The dataset we use has the 10 lowest energy conformations (as computed by the SYBYL software package (www.tripos.com)) for each of 31 thermolysin inhibitors along with their activity levels. The relational background knowledge we have for this data was obtained from David Enot and Ross King and is similar (but not identical) to the background knowledge used in previous work (Marchand-Geneste et al., 2002). This background knowledge defines 26 chemical groups that can be used to define a pharmacophore.

The second dataset we use consists of dopamine agonists (Martin et al., 1993). Dopamine works as a neurotransmitter in the brain, where it plays a major role in movement control. Dopamine agonists are molecules that function like dopamine and produce dopamine-like effects and can potentially be used to treat diseases such as Parkinson's disease. The dataset we use has 23 dopamine agonists along with their activity levels. For this dataset, the number of conformations for each molecule ranges from 5 to 50. The background knowledge we have for this dataset is more limited than in the previous dataset – we know about four groups: hydrogen donors, hydrogen acceptors, hydrophobes and basic nitrogen groups.

The final dataset we use consists of thrombin inhibitors (Cheng et al., 2002). Thrombin works as a blood coagulant and thus its inhibitors can be used as anti-coagulants. The dataset consists of 41 thrombin inhibitors and their activity levels. Each molecule has between 3 and 334 conformations. The background knowledge for this task includes information about six different types of chemical groups.

Experiments. In our experiments, we test three hypotheses. First, we hypothesize that the SAYU procedure results in features that are better suited to regression than a feature construction criterion based on coverage, as standard ALEPH uses. Second, we hypothesize that the MI regression procedure results in more accurate activity predictions than standard linear regression. Third, we hypothesize that the combined MIR-SAYU procedure will yield more accurate predictions than either extension by itself.

To test our hypotheses, we use four baselines along with our algorithm. These are as follows:

1. **Constant**: This algorithm simply predicts the average activity of all molecules in the training set as the activity for every novel molecule.
2. **LR-ALEPH**: This algorithm is the relational approach described in Section 2 and is similar to the framework of Marchand-Geneste *et al.* (2002). It uses ALEPH to construct a set of clauses based on coverage. A single feature vector is generated for each molecule from these clauses. A model is learned using linear regression on these features. For novel molecules, feature vectors are generated using the same clauses and the learned linear model is used to predict activity.
3. **MIR-ALEPH**: This algorithm learns a MI regression model and uses it to predict activity, but uses the standard ALEPH to construct features based on coverage.
4. **LR-SAYU**: This algorithm learns a linear regression model and uses it to predict activity, but uses the SAYU procedure to select features for the model.
5. **MIR-SAYU**: This is our proposed approach, as described in Section 3.

In prior work (Marchand-Geneste et al., 2002), relational approaches have been compared to other activity prediction methods, such as CoMFA (outlined in Section 2), and found to be competitive. Therefore, we restrict our current evaluation to the algorithms mentioned above.

Table 2. Root mean squared errors for different methods on drug activity datasets. Values in bold indicate best results on each dataset. LR refers to linear regression, MIR to multiple-instance regression, and SAYU to the "Score As You Use" rule selection procedure.

Dataset	Constant	LR-ALEPH	MIR-ALEPH	LR-SAYU	MIR-SAYU
Dopamine Agonists	1.38	1.53	1.57	1.25	**0.87**
Thermolysin Inhibitors	1.93	1.47	1.31	1.37	**1.27**
Thrombin Inhibitors	1.56	3.27	1.95	1.36	**1.28**

In our experiments, ALEPH searches over 4-point pharmacophores, as in prior work (Marchand-Geneste et al., 2002). For SAYU, we set the number of internal cross validation folds, n, to be 5 and the r^2 improvement threshold, p, to be 0.2. As a stopping criterion for SAYU, we used a time threshold, allowing each fold one hour of runtime. For MI regression, the softmax parameter α is set to 3 and the regularization factor λ to 1. These parameter values seemed reasonable after some initial exploration; they have not been tuned to these tasks. To optimize our objective functions, we use the L-BFGS algorithm (Fletcher, 1980). To evaluate the algorithms, we use leave-one-molecule-out cross validation. For each dataset, we hold out one molecule in turn as the test molecule and learn a model using the remaining molecules. We then predict the activity of the held-out molecule using the learned model. We report the root-mean-squared (RMS) errors averaged across the held-out molecules in Table 2.

From the table, we observe that for all three tasks, the methods using SAYU outperform the methods that do not use SAYU by a wide margin. In fact, we observe that for both the dopamine and thrombin datasets, LR-ALEPH and MIR-ALEPH both exhibit worse RMSE than the Constant model. This indicates that the coverage measure used by ALEPH to induce features in these domains does not result in features which are able to generalize well to predicting the real-valued activity that we are ultimately interested in. From these results, we conclude that interleaving feature construction and model building using the SAYU procedure results in features that are better able to generalize to predicting activity than features generated by coverage-based measures, such as used by standard ALEPH.

Comparing the two approaches using MI regression to the the ones using linear regression, we observe that in general, the MI approaches outperform their counterparts. The only exception is in the case of dopamine, where MIR-ALEPH is slightly worse than LR-ALEPH. However, we believe this is likely because the features generated by ALEPH are not useful in predicting activity for this case, making any comparison between the linear regression and MI regression difficult. Apart from this case, we observe that MIR-ALEPH is more accurate than LR-ALEPH, and MIR-SAYU is more accurate than LR-SAYU. From these results, we conclude that incorporating knowledge about individual conformations using MI regression generally results in more accurate prediction models than using linear regression on a single feature vector for each molecule.

Finally, we observe that the combined approach we have presented in this work, MIR-SAYU, is the most accurate on all of our 3D-QSAR tasks. It is more accurate than either MIR-ALEPH, which uses MI regression models but does not use SAYU, or LR-SAYU, which uses SAYU, but not MI regression models. From these results, we conclude that combining the two extensions we have presented results in more accurate models than either extension by itself.

SAYU has been shown to consistently produce simpler models than ALEPH (Davis et al., 2005). An interesting question to ask is whether MI regression yields more complex models than linear regression, that is, whether MIR-SAYU learns more complex models than LR-SAYU (the question makes sense only in the context of SAYU, because LR-ALEPH and MIR-ALEPH use the same set of features by design). While we did not enforce any constraint on the total number of features added to each model, we observed that these approaches used approximately the same numbers of features in our experiments. This indicates MIR-SAYU obtains its improvement over LR-SAYU by selecting more informative features (pharmacophores), rather than simply by using more features.

Another interesting question to ask is if the pharmacophores used by our MIR-SAYU models to predict activity have any biological interpretation. In fact, we observed that for dopamine, the rules used by MIR-SAYU on most of the folds agree with the general pharmacophore model in the literature (McGaughey & Mewshaw, 1999). They each have the key basic nitrogen, hydrogen acceptor and hydrophobic group of the model. Since we specified that all rules must encode four-point pharmacophores, the rules all contained either an additional hydrophobic group or hydrogen acceptor; most contained the added hydrophobe. The one exception is a learned pharmacophore that had an extra hydrophobe in the position where the basic nitrogen should be; this feature had a substantial negative coefficient in the regression model. For thermolysin, the known pharmacophore model has seven interaction points, although all seven points are not required for binding. As a result, on every fold of cross-validation multiple rules were learned, capturing different four-point subsets of the seven-point pharmacophore. A combination of such four-point pharmacophores actually makes it possible to achieve better prediction of activity than would be done with a single seven-point pharmacophore, because different coefficients can be attached to each of the four-point pharmacophores. Finally, thrombin is a particularly interesting challenge because no pharmacophore model has been widely agreed upon or validated. Our models are less consistent across folds than for the other tasks, but again our approach shows improved predictive performance. Thus, we conclude that our approach is able to learn models that predict drug activity in terms of biologically meaningful pharmacophores. We expect that this property will prove helpful in analyzing the produced activity models.

5. Conclusion

We have presented MIR-SAYU, a novel machine learning approach for 3D-QSAR. Our approach extends prior work in two ways. First, we use SAYU to construct and select rules that define features, resulting in a tight coupling between feature construction and model building. This permits us to, at any time, learn the rule that most improves our prediction of real-valued activity. Second, we use MI regression for model building. This allows us to separate out the features that are true of each 3D conformer of a molecule. In our experiments on three real-world 3D-QSAR tasks, we observed that each extension by itself improved the accuracy of our predictions. Further, our proposed approach, which uses both extensions, resulted in the most accurate predictions. We also observed that our approach is able to discover biologically relevant pharmacophores when predicting activity. In future work, we plan to explore more complex models of activity prediction, as well as feature construction procedures that search over more complex rule spaces.

Acknowledgements

JD is supported by an NLM training grant to the Computation and Informatics in Biology and Medicine Training Program (NLM 5T15LM007359). VSC was partially supported by CNPq and by Fundação para a Ciência e Tecnologia. DP was supported in part by NSF grant IIS 0534908.

References

Brint, A., & Willett, P. (1987). Algorithms for the identification of three-dimensional maximal common substructures. *J. Chemical Informatics and Computer Sciences*, *27*, 152–158.

Cheng, J., Hatzis, C., Hayashi, H., Krogel, M.-A., Morishita, S., Page, D., & Sese, J. (2002). KDD Cup 2001 report. *SIGKDD Explorations*, *3*, 47–64.

Cramer, R. D., Patterson, D. E., & Bunce, J. D. (1988). Comparative molecular field analysis (ComFA). Effect on binding of steroids to carrier proteins. *Journal of the American Chemical Society*, *110*, 5959–5967.

Davis, J., Burnside, E., Dutra, I. C., Page, D., & Costa, V. S. (2005). An integrated approach to learning Bayesian networks of rules. *Proceedings of the 16th European Conference on Machine Learning* (pp. 84–95). Springer.

Dietterich, T. G., Lathrop, R. H., & Lozano-Perez, T. (1997). Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence*, *89*, 31–71.

Finn, P., Muggleton, S., Page, D., & Srinivasan, A. (1998). Pharmacophore discovery using the inductive logic programming system PROGOL. *Machine Learning*, *30: Special issue on applications and the knowledge discovery process, Kohavi and Provost (Ed.s)*, 241–270.

Fletcher, R. (1980). *Practical methods of optimization*, vol. 1: Unconstrained Optimization, chapter 3. John Wiley and Sons.

Jain, A., Dietterich, T., Lathrop, R., Chapman, D., Critchlow, R., Bauer, B., Webster, T., & Lozano-Pérez, T. (1994a). Compass: a shape-based machine learning tool for drug design. *Journal of Computer-Aided Molecular Design*, *8*, 635–652.

Jain, A., Koile, K., Bauer, B., & Chapman, D. (1994b). Compass: Predicting biological activities from molecular surface properties. *Journal of Medicinal Chemistry*, *37*, 2315–2327.

Landwehr, N., Kersting, K., & Raedt, L. D. (2005). nFOIL: Integrating Naive Bayes and FOIL. *Proceedings of the 20th National Conference on Artificial Intelligence* (pp. 795–800).

Landwehr, N., Passerini, A., Raedt, L. D., & Frasconi, P. (2006). kFOIL: Learning simple relational kernels. *Proceedings of the 21st National Conference on Artificial Intelligence*.

Marchand-Geneste, N., Watson, K., Alsberg, B., & King, R. (2002). New approach to pharmacophore mapping and QSAR analysis using inductive logic programming. Application to thermolysin inhibitors and glycogen phosphorylase *b* inhibitors. *Journal of Medicinal Chemistry*, *45*, 399–409.

Maron, O. (1998). *Learning from ambiguity*. Doctoral dissertation, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA.

Martin, Y., Bures, M., Danaher, E., DeLazzer, J., Lico, I., & Pavlik, P. (1993). A fast new approach to pharmacophore mapping and its application to dopaminergic and benzodiazepine agonists. *J. Computer-Aided Molecular Design*, *7*, 83–102.

McGaughey, G. B., & Mewshaw, R. E. (1999). Application of comparative molecular field analysis to dopamine d2 partial agonists. *Bioorganic Medical Chemistry*, *7*, 2453–2456.

Muggleton, S. (1995). Inverse entailment and Progol. *New Generation Computing*, *13*, 245–286.

Ray, S., & Page, D. (2001). Multiple instance regression. *Proceedings of the 18th International Conference on Machine Learning* (pp. 425–432). Morgan Kaufmann.

Srinivasan, A., Page, D., Camacho, R., & King, R. (2006). Quantitative pharmacophore models with Inductive Logic Programming. *Machine Learning Journal*, *64*, 65–90.

Vapnik, V. (1999). *The nature of statistical learning theory*. Statistics for Engineering and Information Science. Springer.