Active Learning with Support Vector Machines in the Drug Discovery Process[‡]

Manfred K. Warmuth,*[,†] Jun Liao,[†] Gunnar Rätsch,[§] Michael Mathieson,[†] Santosh Putta,[#] and Christian Lemmen[||]

Computer Science Department, University of California, Santa Cruz, California 95064, RSISE, Australian National University, ACT 0200, Canberra, Australia, Rational Discovery LLC, 555 Bryant St. #467, Palo Alto, California 94301, and BioSolveIT GmbH, An der Ziegelei 75, 53757 Sankt Augustin, Germany

Received October 13, 2002

We investigate the following data mining problem from computer-aided drug design: From a large collection of compounds, find those that bind to a target molecule in as few iterations of biochemical testing as possible. In each iteration a comparatively small batch of compounds is screened for binding activity toward this target. We employed the so-called "active learning paradigm" from Machine Learning for selecting the successive batches. Our main selection strategy is based on the *maximum margin hyperplane*—generated by "Support Vector Machines". This hyperplane separates the current set of active from the inactive compounds and has the largest possible distance from any labeled compound. We perform a thorough comparative study of various other selection strategies on data sets provided by DuPont Pharmaceuticals and show that the strategies based on the maximum margin hyperplane clearly outperform the simpler ones.

1. INTRODUCTION

The drug discovery process traditionally involves an iterative procedure of finding compounds that are active against a biological target. Each iteration usually involves selecting or synthesizing compounds from some accessible collections and testing them in a biological assay against the target. Analyzing the resulting data in each round typically provides a better understanding of the reasons for activity, leading to a better design or selection of compounds in the next round of the process.

Computational methods have often been used to aid this process. In each round the data is analyzed and a new preferably interpretable model is constructed that helps select/design compounds for the next round. Note that the design/selection strategy for compounds may differ from round to round. Some of the factors that play a role in this decision include the stage of a project (e.g. early vs late), the available source pool of the compounds (e.g. combinatorial library vs diverse collection), picking new compounds based on already known scaffolds, or finding novel ones (e.g. similar vs diverse). Ideally, the computational methods used to assist this process should be adaptable to these varying requirements. In this paper we discuss several selection strategies that are aimed at addressing some of these issues.

Additionally, we use a rather new paradigm from Machine Learning theory called *active learning*.[2−5] Unlike more conventional learning methods where the data (training set) used to derive the model remains static, we let our data set increment with each round. In each round the algorithms *actively selects* a batch of unlabeled compounds to be tested for activity. Once the results from this batch are known we can label these examples as *active* or *inactive* and *recompute* our model of activity based on all examples labeled so far.

The present approach closely mimics the drug discovery process which is traditionally iterative and where the selection for the next round is commonly based on all the currently available data. We show in this paper that the iterative scheme of active learning can be very powerful even if the underlying model for biological activity is not particularly accurate. (Note that the term active in the expression "active learning" refers to actively exploring the data and rather than biological activity.)

The simplest selection strategy is to choose new compounds at random for testing. Obviously, with random selection, the number of "hits" grows only linearly with the total number of compounds tested. Since most compounds are commonly inactive, this strategy is not very effective. Another simplistic strategy is to pick those compounds that are *closest* to some previously found active compounds. We will show that there are selection strategies based on more sophisticated Machine Learning algorithms that greatly outperform these simple selection strategies in that the total number of actives found in the first few test batches is significantly increased.

Most of the more sophisticated strategies are based on linear models in some high-dimensional descriptor space. (An overview is given in the conference paper.[1]) For the sake of simplicity of presentation we focus on only one such model in this paper. We chose the model which led to one of the best overall selection strategies surveyed in ref 1 and additionally has an easy geometric interpretation. It is based on a particular hyperplane in feature space that separates the

* Corresponding author phone: +1 831 459 4950; e-mail: manfred@cse.ucsc.edu.

[‡] Part of this work has been presented at QSAR Gordon Conference in Tilton, NH, U.S.A. (August 2001). An extended abstract which emphasizes the Machine Learning aspects of our work and compares a large number of selection strategies appeared in the proceedings of the NIPS 2001 conference (see ref 1).

[†] University of California.

[§] RSISE, Australian National University.

[#] Rational Discovery LLC.

[||] BioSolveIT GmbH.

10.1021/ci025620t CCC: $25.00
Published on Web 02/12/2003

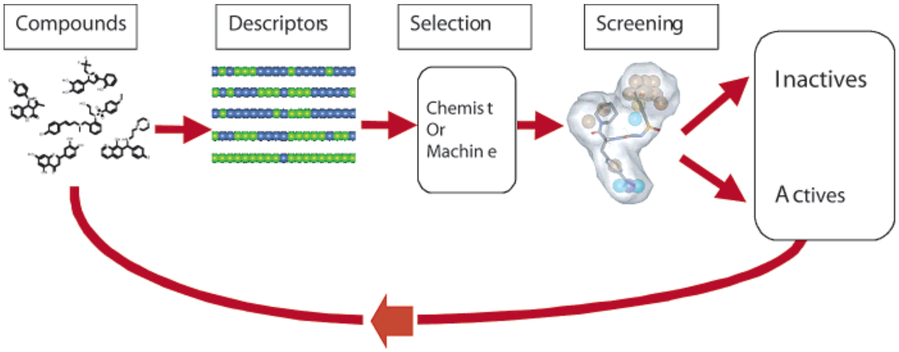

Figure 1. The drug discovery cycle.

active from the inactive compounds called the *maximum margin hyperplane*, i.e., the plane with the largest possible distance from any of the labeled compounds. (Algorithms producing this plane are called "Support Vector Machines".)

Given a separating hyperplane, different selection strategies are possible. One scheme chooses unlabeled compounds that are furthest on the positive side of the current maximum margin hyperplane. We will show that this strategy performs best in terms of picking the most actives in few iterations.

Several machine learning methods have been applied to the drug discovery process to derive a model for activity. Discussing all of them is beyond the scope of this paper. Our main selection strategy is based on the maximum margin hyperplane generated by Support Vector Machines (SVMs;[6−9] see also http://www.kernel-machines.org). SVMs have a number of advantages over classical methods, such as stability, simple geometric interpretation, and use of kernels for nonlinear decisions, and have received much attention in a variety of application fields (for an overview see ref 10). SVMs have been previously employed to model activity of untested compounds.[11−13] However, only a rather small set of conventional real valued descriptors has been used, and the active learning aspect of the problem has not been addressed. One of the data sets we use is the *Thrombin data set* which was the basis of the recent *Knowledge Discovery and Data Mining Competition* (KDD cup, cf. http://www.cs.wisc.edu/∼dpage/kddcup2001). However the setup in this competition with a given training and test set enforced the static type of machine learning. In this work we put a particular emphasis on exploiting the iterative and dynamic nature of the problem.

Almost all machine learning methods rely on a descriptor space that is used to describe each compound in the data set. The compounds are represented in this descriptor space by vectors of descriptor components. As a result, compounds can be thought of points in a *high dimensional descriptor space*. The learning algorithm is not coupled with the descriptor space, and one can take advantage of any advancement made in the descriptor technology. For this study we chose in-house descriptor tools developed at DuPont Pharmaceuticals called the *shape feature and pharmacophore descriptors*.

We start by briefly describing the drug discovery cycle. Next we discuss how selecting test batches naturally becomes a Machine Learning problem. Then Support Vector Machines and various selection strategies are briefly described. Here, we also discuss links to the active learning paradigm and provide motivations for the particular selection strategies. Finally we demonstrate and discuss the performance on two real data sets. We conclude with a summary and discussion of future work.

2. THE DRUG DISCOVERY CYCLE

We are faced with the following situation. A large number of compounds need to be "mined" for finding out quickly which compounds are *active*, i.e., binding to a particular target. The compounds may come from different sources such as vendor catalogs, corporate collections, or combinatorial chemistry. In fact, the compounds need to exist only virtually, being defined in terms of their *descriptor* vectors (cf. Section 5.1).

In the *drug discovery cycle* (cf. Figure 1)[14] one typically starts with some initial set of already tested compounds. Then the chemists iteratively design/select batches of compounds for testing. Note that it is more efficient to test multiple compounds in parallel. However, often only a small number of chemical classes can be pursued in parallel. The idea is to refine the model of activity in each step, based on all tested compounds at hand and to choose the most promising compounds for the next batch. The cycle is repeated until the ultimate goal is achieved, i.e., active compounds with good enough properties for a clinical trial are found.

In this paper we attempt to do the selection step in the cycle by the aid of a Machine Learning algorithm. At any stage of the process, three types of compounds can be distinguished: (a) a very small fraction of compounds that already have been identified as active, (b) a much larger fraction of compounds that already have been identified as inactive, and (c) by far the largest fraction of compounds that have not yet been tested (the *unlabeled* compounds). This situation is illustrated in Figure 2, where for the sake of simplicity the descriptors have only two components (so we obtain a two-dimensional plot).

It is important to note that our Machine Learning algorithm does its selection based on all previous test batches. Such learning approaches are collectively called *active learning techniques*. As we shall see in the experiments, if we restrict the algorithm not to make use of the cumulative information from previous batches, i.e., only consider a static setup, then the performance degrades dramatically.

3. ACTIVE LEARNING WITH SVMS

There are many Machine Learning techniques to choose from. We considered a few of them in an earlier study.[1] All of these are based on an essentially linear model of activity

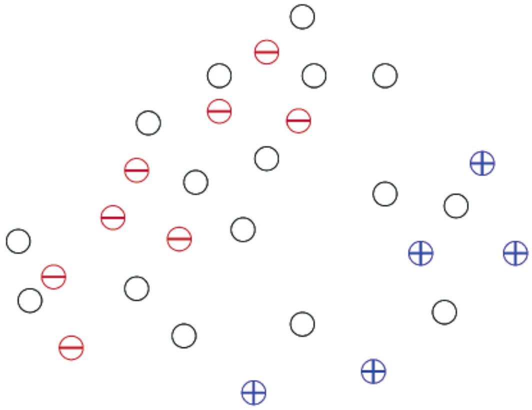

Figure 2. Three types of compounds/points in a (hypothetical) two-dimensional descriptor space: ⊕ (blue) are active, ⊖ (red) are inactive, and ○ are yet unlabeled.

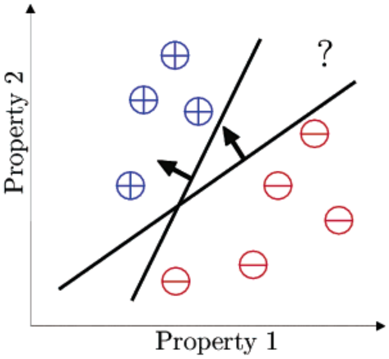

Figure 3. Linear separation. There are many hyperplanes that could separate the data.

(some of the methods combine several linear predictors). In this paper we describe a particular one based on Support Vector Machines (SVMs). We chose SVMs for our presentation, because they have a simple geometric motivation and also yield very good results. Note however, other algorithms from ref 1 may well be used instead of SVMs, and the discussion listed below about selection strategies holds for them as well.

By a linear model we mean a *hyperplane* that divides the descriptor-space into two parts. With a *separating* hyperplane, all known active compounds lie in the positive half-space and all known inactive compounds in the negative half-space. (Points **x** on a hyperplane satisfy the equation $\mathbf{w}\cdot\mathbf{x} + b = 0$, for some weight vector $\mathbf{w}\in\mathbf{R}^n$ and bias $b\in\mathbf{R}$. For points **x** in the positive half-space, $\mathbf{w}\cdot\mathbf{x} + b > 0$.) For a given hyperplane, the score of a compound is the signed distance to the hyperplane. (The *signed distance* of point **x** to plane ($\mathbf{w}$, b) is defined as $(\mathbf{w}\cdot\mathbf{x}+b)/(||\mathbf{w}||_2)$, and the *distance* is the absolute value of this quantity.) So the compounds on the plane have score zero, the compounds in the negative half-space all have a negative score, and the rest a positive score.

In fact a variety of different hyperplanes may separate the data correctly (see Figure 3). Support Vector Machines choose a particular separating hyperplane—the so-called the *maximum margin hyperplane* (see Figure 4). The *margin* of a separating hyperplane is the minimum distance of any labeled data point to the hyperplane. Intuitively speaking, the larger the margin the clearer the separation between the known actives and known inactives; hence, it is natural to choose the maximum margin hyperplane as a robust classifier. Furthermore, results from Statistical Learning Theory show that this approach is sensible, if the data are separable. Usually only a small set of vectors called *support vectors* line up closest to the decision boundary (i.e. the value of their distance to the boundary equals the margin).

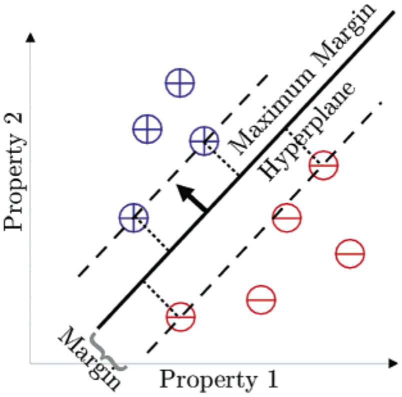

Figure 4. Maximum margin hyperplane. The minimum distance of the examples to the hyperplane is maximized.

In general, linear functions might be suspected to be insufficient to solve complex classification tasks. However, this depends on the representation of the data points. The descriptors used in our study, though sparse, are very large. With such a high number of dimensions, linear classifiers were found to be sufficient to separate the data. Moreover, SVMs do allow for introducing nonlinearity by projecting the descriptor vectors into a feature-space of arbitrary dimension. By making use of a *kernel* this projection can be achieved without computational overhead (cf. ref 6). We tried various expansions of the descriptor features (results not shown), but this did not lead to an improved performance. It seems that the descriptor vectors developed by the computational chemists already contain a rather complete set of features.

Most Machine Learning algorithms are designed for the static setting, where one is given a few labeled examples and asked for predictions on unseen examples without intermediate feedback from testing. A fairly new research direction in Machine Learning—called *Active Learning*—addresses exactly this issue.[2,15,16,5] The considered algorithms iteratively select examples from a pool that improve the internal model as quickly as possible. The active learning approach matches exactly the drug discovery cycle. Also, SVMs were found to be very suitable for the active learning setup (cf. refs 5 and 17).

4. SELECTION STRATEGIES

Probably the simplest selection strategy is to choose the next batch at *random* from the unlabeled compounds. This strategy does not make use of the labels obtained in previous iterations. The number of active hits grows only linearly with the number of iterations. Since the number of actives is usually quite small, the performance of the *random selection* strategy is poor.

Another straightforward selection strategy is to pick unlabeled compounds that are *closest to previously known actives*. Different distance measures on binary descriptors are possible here. We used the total number of bits differing in the two vectors as a distance measure. An unlabeled

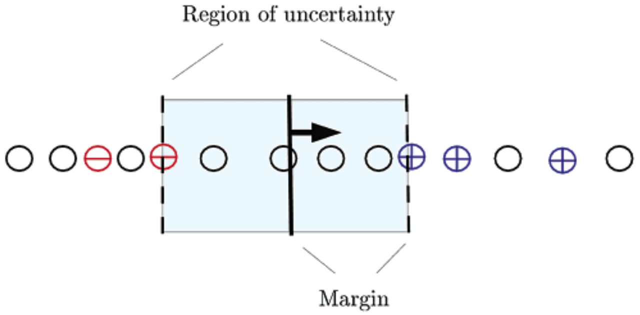

Figure 5. Binary search in one-dimensional case: Select unlabeled points *closest to plane*.

compound receives the (negative) *distance to its closest active compound* as a score. The strategy then is to pick those unlabeled compounds with the highest scores (i.e. smallest distance to another active). Note that this strategy takes actives into account that are found in previous iterations. However, it searches only locally and will not find actives that are remote from the previously known ones.

A more sophisticated model of activity is the linear one derived by SVMs. As mentioned above they compute a *score* of activity (signed distance to the hyperplane) that is used for prediction or selection.

Given the model for activity, the obvious selection strategy is to select the compounds with *largest positive* score, since they are most likely to be active. We shall see that this is a good strategy to find many compounds in a few iterations, which is clearly one of the primary goals in drug discovery.

If, however, the goal is to understand the structure–activity relationship, then it is most important to rapidly improve the model. We will show that in this case the best strategy is to select examples near the decision boundary. The effectiveness of the *near boundary* selection strategy shall be motivated using a one-dimensional illustration (cf. Figure 5).

Note that we assume linear separability of the compounds (which is possible with our high-dimensional descriptors). In the one-dimensional case illustrated in Figure 5 this means that going from right to left there is a sequence of actives until the leftmost active is reached. Going further to the left we run into the rightmost inactive followed by all inactives to the left. To determine the boundary between “active” and “inactive”, it is most effective to test unlabeled compounds that are near the boundary and of distance less than the margin away from the boundary. Independent from the result of such a test, i.e., the activity of the respective compound, the area of uncertainty will be reduced by almost a factor of 2. The strategy is similar to a binary search and suggests *exponential* convergence to the optimal classifier. This type of argument can be generalized to arbitrary dimensions utilizing the so-called *version-space*, whose volume decreases exponentially. For details on the theoretical justification see ref 1.

Figure 5 can also be used to illustrate the difference between the *largest positive* and *near boundary* selection strategies. One of the four unlabeled compounds within the margin is the rightmost active. It takes at most three tests (i.e. $\geq (\log_2 4) + 1$) to determine the leftmost active using the near boundary selection. Hence, the model of activity is determined quickly with the *near boundary* selection strategy. The *largest positive* strategy tests from right to left. It would take up to seven tests (in the worst case) to determine the leftmost active instead of three; however, this procedure uncovers many active compounds already along the way.

These observations suggest that the iterative refinement of the model is an essential part of any effective selection strategy. We will provide experimental evidence for this in the next section. Note that the SVM only models whether a compound is active (i.e. above some activity threshold) or inactive (i.e. below the threshold), but not its activity level which would be a harder task. Hence, the distance of a compound to decision hyperplane is not necessarily related to the strength of binding and highly active compounds can be close to the boundary. So there is no reason to believe that the *nearest to boundary* selection scheme will by definition pick the examples with low activity which, of course, would be of less value. Empirically we confirmed for our data sets (results not shown) that the activity of an active compound is uncorrelated with its distance to the hyperplane.

5. EXPERIMENTS

5.1. Data Sets. Our experiments are based on a data set provided by Dupont Pharmaceuticals for which Thrombin was the target. This data set was also used for a recent competition, the Knowledge Discovery and Data mining Cup 2001 (cf. http://www.cs.wisc.edu/∼dpage/kddcup2001). We extensively tested our algorithms in a second much larger internal data set with CDK2 as the target. The results were similar, and for simplicity we only report the results on Thrombin.

The Thrombin data consists of two *rounds* of data. Round_0 is the result of an initial screen against CombiChem’s *Universal Informer Library* (UIL).[18] This is a diverse collection of compounds routinely used for target validation and initial screening. The entire UIL has been reduced here to the subset of compounds that contain a positive charge which is a known predominant feature in Thrombin actives. Additionally, a number of literature active compounds have been included in round_0. Round_1 is the result of an informative library design around five templates, based on the medicinal chemistry insight gained from the round_0 data. Thus, round_1 is already a highly enriched data set.

After removing 593 compounds that only had zero entries in all descriptor components, round_0 consists of 1316 compounds with 40 nominated actives. Round_1 has 634 compounds with a total of 150 actives. Each descriptor vector has 139 351 binary components. The average number of nonzero bits is 1378 in round_0, 7613 in round_1. The descriptors were produced by internal software tools developed at DuPont Pharmaceuticals for shape-based comparison and alignment of compounds (see refs 19 and 20).

The CDK2 data set has a similar history: 14 223 compounds were tested in the initial UIL screen and constitute, enriched with literature data, round_0 of this data set. In successive rounds of synthesis and design, the medicinal chemists made 3232 additional compounds following the paradigm of the drug discovery circle. There were 255 actives in round_0 and 108 actives in the later rounds. The descriptors were produced by internal software tools developed at DuPont Pharmaceuticals for generating traditional pharmacophore descriptors (see ref 21 for details). The dimensional-

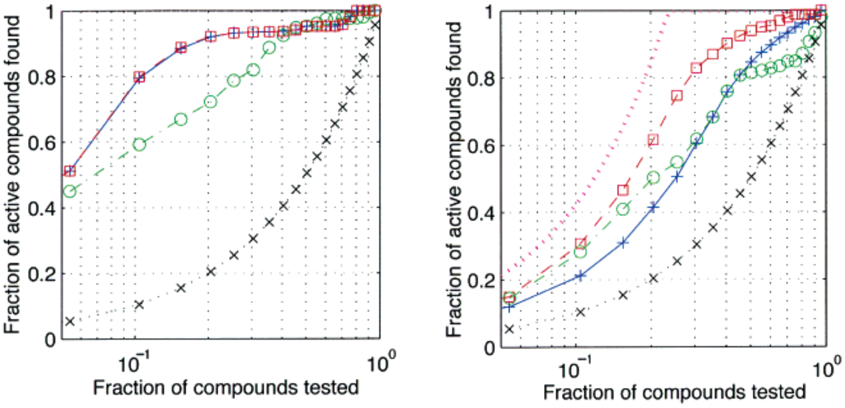

Figure 6. We plot the total fraction of hits (in 5% test batches) for round_0 (left) and round_1 (right) of the Thrombin data set. In each case we plot all four selection strategies as a function of the fraction of compounds tested: random (black "x"), *closest to an active* (green circle), *largest positive* (red box), and *near boundary* (blue plus). For round_0, the total number of actives is less than 5%. For round_1 the magenta curve shows the *optimal strategy* which picks only actives in each test batch until all actives are selected.

ity of the descriptor vectors is up to 35 926 557, with an average of 78 602 nonzeros descriptor components in the combined data set.

5.2. Comparison of Selection Strategies. A selection procedure is specified by three parameters: initialization, batch size, and selection strategy. In practice it is not cost-effective to test a single example at a time. For most of the paper we fixed the batch size to 5% of the total number of unlabeled compounds in the data set, which appears reasonable in comparison with typical experimental constraints. Moreover, one obtains only negligibly more active hits when testing a single example in each round instead of 5% batches (result not shown, cf. ref 1).

We used the following initialization: Initial batches from round_0 or round_1 are chosen at random until at least one active and one inactive example is found. Typically this was achieved already with the first 5% batch. All subsequent batches are then chosen using the selection strategy.

In Figure 6 we plot the total fraction of hits (in the test batches) for all four methods: *random*, *closest to an active*, *SVM largest positive*, and *SVM near boundary*. To provide an upper bound we also plot the number of hits of the unrealistic *optimal* selection strategy which chooses purely active compounds in the test batches until all active compounds have been selected. The fraction of hits of the random selection strategy grows linearly with the fraction of examples tested. *Closest to an active* is inferior because it does a local search. *Largest positive* is closest to the *optimum selection*. *Near boundary* performs not as well as the *largest positive* strategy but is not much worse.

In Figure 6 we report the averages of 10 runs. (Each run is initialized with a different random batch.) For all SVM-based selection strategies reported in this paper, we always normalize the descriptor vectors by their two-norm. This normalization consistently improves the performance (not shown). We use SVM-light[22] for our SVM implementation.

5.3. Active vs Passive Learning. An interesting question arises: Where does this good performance come from? It could either be that the model of activity is very good and seeing only a few examples leads to very good predictions on the rest of the examples or it could be that the iterative (i.e. active) learning scheme is of prior importance.

To investigate this, we performed the following experiment. We tried a different way of initialization where the SVM is first trained on all round_0 data. Then, based on the resulting model we select batches of 5% from the round_1 data. In one case we keep on refining the model with each new batch, and in the other case we stick to the initial model that was built based on the round_0 data. In both cases we select the examples with the largest positive score.

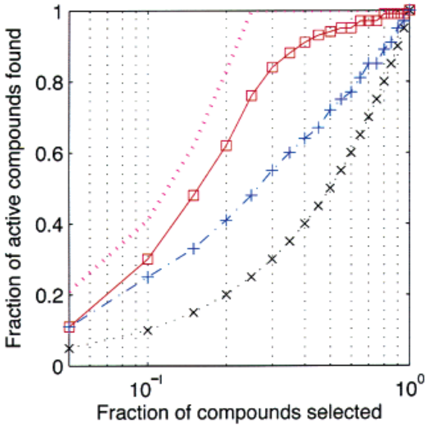

Figure 7. Active vs passive learning: Shown is the fraction of hits obtained by training the SVM on Thrombin round_0 and then iteratively selecting test batches of examples using the *largest positive* selection strategy from Thrombin round_1. In one case we use all recently tested compounds (red square) to update the maximum margin hyperplane. In the other case the initial plane (based on round_0) is kept fixed (blue plus). *Random* (black "x") and *Optimal* (dotted magenta) are shown as references.

The number of hits for both methods on the Thrombin data set is shown in Figure 7. We observe that the number of hits achieved by the active learning strategy is much higher than with the passive strategy. This is consistent for all iterations. For instance, after testing 30% of the data, Active Learning found 84%, whereas the other found only 55% of all actives. We believe that this constitutes convincing experimental evidence that the iterative refinement of the model should be an essential part of any effective selection strategy.

5.4. Exploration vs Exploitation. In Figure 8 we see that the *near boundary* strategy is better at "exploration" (i.e., giving better generalization on the entire data set) while the *largest positive* strategy is better at "exploitation" (i.e., higher number of total hits). One might actually switch between strategies at different stages of a project. In the lead evolution

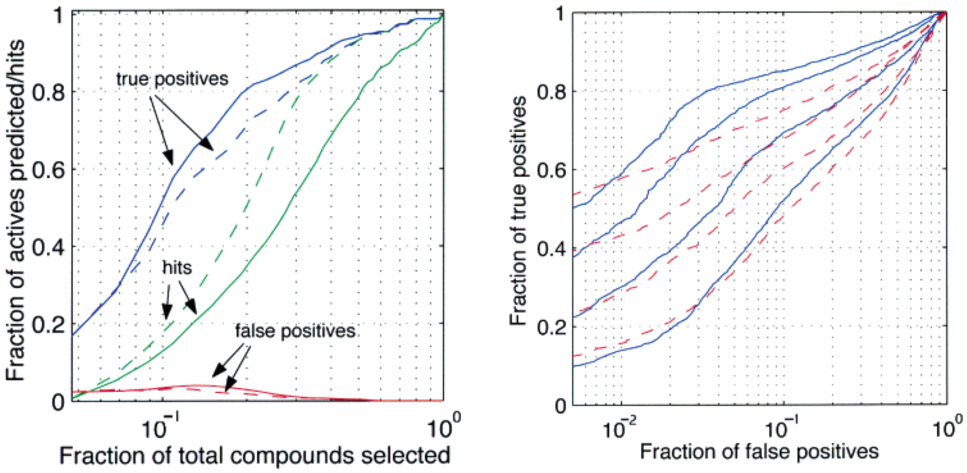

Figure 8. Exploitation versus exploration: (left) Total hits performance (exploit) and true and false positives performance on the whole set (explore) and (right) ROC plots of the classifiers after the selecting the second, fourth, sixth and eighth batch. The dashed line shows the performance of the *largest positive* strategy and the solid line the performance of the *near boundary* method. We used Thrombin round$_1$ data and a batch size of 2% (13 compounds). The initialization batch is random 2% of Thrombin round$_1$.

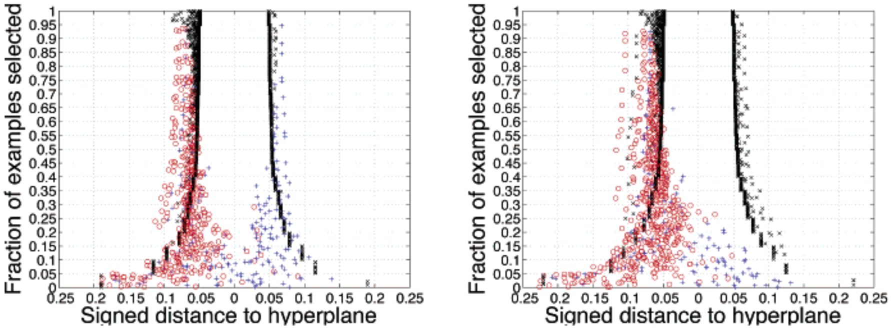

Figure 9. Scatter plot of the signed distance of examples to the hyperplane: (Left) *near boundary* selection strategy, (right) *largest positive* selection strategy. Each stripe shows location of points after an additional 5% batch has been labeled (using SVM). Selected examples are black, unselected actives are blue, unselected inactives are red. We used Thrombin round$_1$. The initialization batch is random 5% of Thrombin round$_1$.

phase one needs to find actives quickly, whereas in lead optimization, a more refined model of activity is required. At the latter stage, chemists know already how to make actives but need to understand in detail which are the important factors for binding to the target.

We gave a simple one-dimensional motivation of our selection strategies using Figure 5. However the dimension of our data is around 139 351. Using a trick we can obtain a one-dimensional snapshot of our partially labeled data by projecting each example onto the normal direction of the current maximum margin hyperplane. Thus each example maps to a signed distance to the hyperplane. In Figure 9 we visualize the location of all examples after each 5% test batch and use different colors for the already selected and unselected examples of each label. To show the density of each type of examples along the normal direction we scatter the points within a thin stripe. Note that each stripe corresponds to a differently oriented hyperplane in the descriptor space and the hyperplane crosses each stripe at the zero-point.

The "minimum margin" can be seen as the margin of the selected examples (black in the plot) that are the closest to the center. The left plot shows the progress of the *closest to boundary* selection strategy and the right plot the *largest positive* selection strategy. During the initial batches, the minimum margin shrinks quickly and then stabilizes. The minimum margin of the left plot shrinks a little bit faster because the *closest selection* strategy stresses exploration. As soon as the "window" between the support vectors is cleaned (at around 50%), the label of most examples is predicted correctly. As we see in Figure 8 (left), after 50% of the examples are selected, 93% of the actives and almost all of the inactives are predicted correctly.

Again we want to point out that we only report results on the Thrombin data set in this paper. However, our algorithms achieved similar performance on the larger CDK2 data set.

6. DISCUSSION AND SUMMARY

The present work discusses the application of Support Vector Machines to the problem of finding active compounds at different stages of the drug discovery process. In therapeutic projects this selection is commonly done during several rounds of design with the goal of finding active compounds within multiple lead series quickly and in parallel developing a more and more refined model of activity. In this scenario we need Machine Learning methods that build the best model based on all currently available labeled data and use this model for suggesting the most critical compounds to be tested next.

In Machine Learning research, the requirements described above are met by active learning methods. Support Vector Machines have been successfully applied to various active learning problems, and in the present work they show convincing results in the domain of computer-aided drug design. For this particular application, we found that the total number of active hits versus the total fraction of selected examples is the most useful performance plot. We also found that selecting the unlabeled examples that are furthest on the positive side of the maximum margin hyperplane leads to the best performance.

A number of additional selection strategies based on other Machine Learning techniques (such as the Voted Perceptron and the Bayes Point Machine) are discussed in the earlier study.[1] They perform similar to the SVM results quoted in this paper. For each of those methods the *near boundary* and *largest positive* selection strategies are available. Generally the *largest positive* strategies produced the most actives early on. We also compared these selection strategies to the *k*-Nearest Neighbor algorithm. Choosing *k* around 20 led to the best performance. However, SVM and the other methods were clearly superior (results not shown).

In the authors' opinion the most important follow-up research on the work presented here comprises two major areas. The first is the reduction of the descriptor space. The maximum margin hyperplane remains unchanged if all examples except the support vectors are removed. While keeping only the support vectors reduces the number of data points (to 16% in Thrombin round_0, 53% in Thrombin round_1), the dimensionality of the descriptor vectors remains unchanged at 139 351. Previous work (in the off-line setting) has shown that for the Thrombin data set, about 40 descriptor components are relevant for the discrimination of actives from inactives.[13] Thus it is desirable to have a method at hand that simultaneously improves the classifier for the purpose of iteratively selecting good test batches and at the same time does this based on a small number of descriptor components. Second, the selection strategies presented here leave room for secondary selection criteria. For example we would like to select a set of compounds that is far on the positive side of the maximum margin hyperplane but also chemically *diverse*.

ACKNOWLEDGMENT

M. K. Warmuth's and Jun Liao's work has been partially supported by the NSF (CCR 9821087) and by a generous gift from Dupont Pharmaceuticals. G. Rätsch's work has been partially supported by the DFG (JA 379/9-1, MU 987/1-1) and travel grants from EU (NeuroColt II). Parts of this work was done while Gunnar Rätsch was at Fraunhofer Institute FIRST, Berlin and at University of California at Santa Cruz.

REFERENCES AND NOTES

(1) Warmuth, M. K.; Rätsch, G.; Mathieson, M.; Liao, J.; Lemmen, C. Active learning in the drug discovery process. In *Adv. in Neural Inf. Proc. Sys. 14*; Dietterich, T. G., Becker, S., Ghahramani, Z., Eds.; MIT Press: Cambridge, MA, 2002; pp 1449−1456,.

(2) Angluin, D. *Machine Learning* **1988**, *2*, 319−342.

(3) Atlas, L.; Cohn, D.; Ladner, R.; El-Sharkawi, M. A.; Marks, R. J.; Aggoune, M. E.; Park, D. C. Training connectionist networks with queries and selective sampling. In *Adv. in Neural inf. proc. sys. 2*; Touretzky, D., Ed.; Morgan-Kaufmann, 1990; pp 566−573.

(4) Bachrach, R.; Fine, S.; Shamir, E. Learning using query by committee, linear separation and random walks. In *Proc. Eurocolt'99*; Springer: Heidelberg, 2000; Vol. 1572 of *LNAI*, pp 34−49.

(5) Campbell, C.; Cristianini, N.; Smola, A. Query learning with large margin classifiers. In *Proc. ICML2000*; Stanford, CA, 2000; p 8.

(6) Boser, B. E.; Guyon, I. M.; Vapnik, V. N. A training algorithm for optimal margin classifiers. In *Proc. ACM Workshop on Computational Learning Theory*; Haussler, D., Ed.; 1992; pp 144−152.

(7) Vapnik, V. N. *The nature of statistical learning theory*; Springer-Verlag: New York, 1995.

(8) Burges, C. J. C. *Knowledge Discovery Data Mining* **1998**, *2*(2), 121−167.

(9) Schölkopf, B.; Smola, A. J. *Learning with Kernels*; MIT Press: Cambridge, MA, 2002.

(10) Müller, K.-R.; Mika, S.; Rätsch, G.; Tsuda, K.; Schölkopf, B. *IEEE Trans. Neural Networks* **2001**, *12*(2), 181−201.

(11) Burbidge, R.; Trotter, M.; Buxton, B.; Holden, S. *Comput. Chem.* **2001**, *26*(1), 4−15.

(12) Rätsch, G.; Demiriz, A.; Bennett, K. *Machine Learning* **2002**, *48*(1−3), 193−221.

(13) Weston, J.; Pérez-Cruz, F.; Bousquet, O.; Chapelle, O.; Elisseeff, A.; Schölkopf, B. Feature selection and transduction for prediction of molecular bioactivity for drug design. Submitted to *Bioinf.* **2002**.

(14) Myers, P.; Greene, J.; Saunders, J.; Teig, S. *Today's Chemist at Work* **1997**, *6*, 46−53.

(15) Cohn, D.; Ghahramani, Z.; Jordon, M. I. Active learning with statistical models. In *Advances in Neural information processings systems*; MIT Press: 1995; Vol. 7, pp 705−712.

(16) Sollich, P.; Saad, D. Learning from queries for maximum information gain in imperfectly learnable problems. In *Adv. in Neural Inf. Proc. Sys. 7*, MIT Press: 1995; pp 287−294.

(17) Tong, S.; Koller, D. Support vector machine active learning with applications to text classification. In *Proc. 7th Int. Conf. Mach. Learning*; Morgan Kaufmann: San Francisco, CA, 2000.

(18) Saunders, J.; Myers, P. L.; Barnum, D.; Greene, J. W.; Teig, S. L. *Genetic Eng. News* **1997**, *17*, 35−36.

(19) Lemmen, C.; Molecular superpositioning − a powerful tool for drug design. In *Proc. 13th European Symposium on QSAR: Rational Approaches to Drug Design*; Prous Science: 2000.

(20) Putta, S.; Lemmen, C.; Beroza, P.; Greene, J. *J. Chem. Inf. Comput. Sci.* **2002**, *42*(5), 1230−1240.

(21) Eksterowicz, J. E.; Evensen, E.; Lemmen, C.; Brady, G. P.; Lanctot, J. K.; Bradley, E. K.; Saiah, E.; Robinson, L. A.; Grootenhuis, P. D. J.; Blaney, J. M. *J. Molecular Graphics Modelling* **2002**, *20*(6), 469−477.

(22) Joachims, T. Making large-scale SVM learning practical. In *Advances in Kernel Methods - - Support Vector Learning*; Schölkopf, B., Burges, C. J. C., Smola, A. J., Eds.; MIT Press: Cambridge, MA, 1999; pp 169−184.

CI025620T