# Towards Optimal Descriptor Subset Selection with Support Vector Machines in Classification and Regression

*Holger Fröhlich*, Jörg K. Wegner, Andreas Zell*

Center for Bioinformatics Tübingen (ZBIT), Sand 1, 72076 Tübingen, Germany

{holger.froehlich,joerg.wegner,andreas.zell}@informatik.uni-tuebingen.de

**Keywords:** Descriptor Selection, Support Vector Machines, Human Intestinal Absorption, aqueous solubility

**Abbreviations:** HIA, Human Intestinal Absorption; SVM, Support Vector Machine; SVR, Support Vector Regression; RFE, Recursive Feature Elimination; IRRM, Incremental Regularized Risk Minimization

**Received on**

**Abstract**. In this paper we present a novel method for selecting descriptor subsets by means of Support Vector Machines in classification and regression – the Incremental Regularized Risk Minimization (**IRRM**) algorithm. In contrast to many other wrapper methods it is fully deterministic and computationally efficient. We compare our method to existing algorithms and present results on a Human Intestinal Absorption (HIA) classification data set and the Huuskonen regression data set for aqueous solubility.

## 1 Introduction

Selecting optimal descriptors to represent molecules in the chemical space is a critical and important step, especially if one is interested in QSAR studies. It has been shown [2, 11, 15, 19] that the quality of the inferred model strongly depends on the selected molecular descriptors. The question is, however, how one can select "good" descriptors. Often this is done in a very heuristic way by experience or taking an educated guess. The first problem with this approach is, that in general one cannot assume that a descriptor which is good for one dataset will be automatically good for another dataset as well (otherwise there would exist a universally best set of descriptors for all problems – which is a contradiction to the *No Free Lunch Theorem* [33, 34]). The second problem is, that even if we have a candidate set of promising descriptors, some descriptors can contribute more to the model than others, and it is possible that irrelevant descriptors degrade the performance of our model. Hence there is the question how we can systematically select a subset of our original descriptors which is suited best for the model we want to infer.

In the following we will view descriptors as *features* of some data $D = (\boldsymbol{x}_1, y_1), ..., (\boldsymbol{x}_n, y_n) \in X \times Y$ which are drawn independently and identically distributed from some unknown (but fixed) probability distribution $P(\boldsymbol{x}, y)$. $X$ is assumed to be some

vector space of dimension $d$ and $Y$ a nonempty set of outputs. From our data $D$ we want to infer a model $f$: $X \rightarrow Y$ about underlying regularities of the data. That means we are considering the situation of supervised learning. In the first step we will focus on the case where we want to learn a classification $f$: $X \rightarrow \{+1,-1\}$. Later on we will also consider regression functions $f: X \rightarrow \Re$.

From the original $d$ descriptors we want to select a subset of $m \ll d$ descriptors such that our classifier $f$ yields the smallest expected generalization error [32]. In other words, we are looking for a subset of $m$ features that discriminate our data best.

The two main approaches to deal with the descriptor selection problem are the filter and the wrapper approach [2, 11, 15, 17]: In a filter method descriptor selection is performed as a preprocessing step to the actual learning algorithm, i.e. before applying the classifier to the selected descriptor subset. Descriptors are selected with regard to some predefined relevance measure which is independent of the actual generalization performance of the learning algorithm. This can mislead the selection algorithm. Wrapper methods, on the other hand, train the classifier with a given descriptor subset as an input and return the estimated generalization performance of the learning machine as an evaluation of the descriptor subset. This step is repeated for each subset taken into consideration.

Unfortunately, for an original set of $d$ descriptors there are $\binom{d}{m}$ different descriptor subsets of size $m$. Hence for large values of $d$, like it is the case in QSAR studies, it is practically impossible to evaluate all these subsets. Therefore, heuristics have to be used to solve this problem approximately. Often used heuristics are backward elimination and forward selection [2, 10, 16], but also stochastic search methods like Genetic Algorithms [8, 22, 24, 27, 28]. However, a major drawback of Genetic Algorithms is their nondeterministic character, which depends on the seeds of the random number generator and makes it difficult to reproduce

solutions obtained by these algorithms. In this paper we will concentrate on deterministic methods.

In the next section we will first review the general problem of model induction in machine learning and its link to Support Vector Machines (SVMs) [6]. We will see the strong connection between general principles in Machine Learning and the problem of descriptor selection. After these theoretical foundations we will present some existing wrapper methods for descriptor selection with SVMs as well as the well known mutual information based descriptor selection as an example of a filter approach [3]. Next we will explain our own algorithm, and we will show the results obtained by this method on a Human Intestinal Absorption data set [30]. After this we will slightly generalize our algorithm to deal with regression problems, and we will show our results on the Huuskonen data set for aqueous solubility [13].

## 2 Descriptor Selection and Machine Learning

A goal of every model $f$ is to minimize the expected generalization error (or *risk*) over all possible patterns which are drawn from the unknown probability distribution $P(\boldsymbol{x}, y)$ [25, 26]

$$R[f] = \int_{X \times Y} \ell(y, f(\boldsymbol{x}))dP(\boldsymbol{x}, y) \quad (1)$$

with $\ell : X \times Y \to \Re$ being some loss-function, i.e. a function which measures the error we make, if our model $f$ predicts $f(\boldsymbol{x})$, but the correct answer would be $y$. In case of a classifier we usually have

$$\ell(y, f(\boldsymbol{x})) = \begin{cases} 1 & \text{if } f(\boldsymbol{x}) \neq y \\ 0 & \text{otherwise} \end{cases} \quad (2).$$

However, we cannot compute $R$ as we do not know $P$. On the other hand it is a crucial insight of Statistical Learning Theory [25, 26] that minimizing the *empirical risk* (or training error)

$$R_{\text{emp}}[f] = \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, f(\boldsymbol{x}_i)) \qquad (3)$$

does not guarantee a minimum of $R[f]$. Thus instead we should minimize the *regularized risk*

$$R_{\text{reg}}[f] = R_{\text{emp}}[f] + \lambda\Omega[f] \qquad (4)$$

which is an upper bound on $R[f]$. The term $\Omega[f]$ is a measure for the *capacity* (which can be thought of as the models' complexity) of the classifier and $\lambda > 0$ is a constant which regularizes the trade-off between training error minimization and reduction of the model complexity.

In this paper we consider SVMs as the learning machine. The usefulness of SVMs in drug design has for instance recently been shown by Byvatov et al. [4, 5]. In SVMs one usually chooses $\Omega[f] = ½||\boldsymbol{w}||^2$ where $\boldsymbol{w}$ is the weight vector of the separating hyperplane in feature space $\mathcal{H}$ and $\|\cdot\|$ denotes the Euclidian norm. It is also worth mentioning that $||\boldsymbol{w}||$ is inverse to the size of the margin between the two classes +1 and –1. Hence, SVMs exactly implement the idea of regularized risk minimization by maximizing the margin between the two classes. As well known this is achieved by solving the quadratic program [6, 23]

$$\begin{aligned} &\min_{\boldsymbol{w}} \frac{1}{2}\|\boldsymbol{w}\|^2 + \mathrm{C}\frac{1}{n}\sum_{i=1}^{n}\xi_i \\ &\text{subject to } y_i(\langle \boldsymbol{w}, \phi(\boldsymbol{x}_i)\rangle + b) \geq 1 - \xi_i \end{aligned} \qquad (5)$$

where $\phi : X \to \mathcal{H}$ is a (possibly nonlinear) map of the original data into some Hilbert space $\mathcal{H}$. This can be equivalently formulated in its dual form

$$\begin{aligned} &\min_{\boldsymbol{\alpha}} W^2(\boldsymbol{\alpha}) = \sum_{i,j=1}^{n} \alpha_i\alpha_j y_i y_j k(\boldsymbol{x}_i, \boldsymbol{x}_j) - \frac{1}{2}\sum_{i=1}^{n}\alpha_i \\ &\text{subject to } 0 \leq \alpha_i \leq \mathrm{C},\ i = 1,...,n, \text{ and } \sum_{i=1}^{n}\alpha_i y_i = 0 \end{aligned} \qquad (6)$$

where $k : X \times X \to \mathcal{H}$ is a kernel function and C a constant that regularizes the trade-off between training error minimization and margin maximization.

To perform descriptor selection we wish to minimize our regularized risk. Hence we should increase the margin between classes +1 and –1. In this way descriptor selection can be viewed as controlling the capacity of the classifier.

# 3 Related Methods

## *3.1 SVM Wrapper Algorithms*

On way of performing descriptor selection by minimizing the regularized risk is the Recursive Feature Elimination (RFE) algorithm of I. Guyon et al. [11]: Let $\boldsymbol{\alpha}^*$ be the solution of (6) with regard to the current descriptor subset, i.e. after training a SVM with the current descriptor subset. Let $\boldsymbol{x}^{(-t)}$ denote that descriptor $t$ has been removed from the molecule $\boldsymbol{x}$. Assuming that the set of support vectors does not change significantly when eliminating just one descriptor, RFE removes the $r$ descriptors for which the change in the regularized risk

$$DW^2 = \left| \sum_{i,j=1}^{n} \alpha_i^* \alpha_j^* y_i y_j k(\boldsymbol{x}_i, \boldsymbol{x}_j) - \sum_{i,j=1}^{n} \alpha_i^* \alpha_j^* y_i y_j k(\boldsymbol{x}_i^{(-t)}, \boldsymbol{x}_j^{(-t)}) \right| \qquad (7)$$

($t$ = 1,…,d) is smallest. Usually $r$ is set to half of the number of existing descriptors. The procedure begins with the set of all descriptors and is repeated until the desired number $m$ of descriptors has been reached. As an output of the algorithm we receive a ranking of all descriptors according to the time of their removal and the measure $DW^2$.

Alternatively to removing the $r$ descriptors for which (7) is smallest, one could also use a gradient based strategy, i.e. compute the derivative of $W^2$ in (6) with respect to some continues scaling factor $\theta_t \in \Re$ $(t = 1,...,d)$ for each descriptor. This approach has been investigated recently e.g. in [21] and [5].

Another method for SVMs is the $\ell_2$-AROM algorithm of J. Weston et el. [31]. Here the idea is to successively eliminate descriptors for which the components of the hyperplane normal

vector $\boldsymbol{w}$ are relatively small. However, this method usually cannot handle nonlinear problems.

### *3.2 A Filter Algorithm – the Mutual Information Measure*

A classical way to perform descriptor selection with a filter is to keep only those $m$ descriptors $t$ for which the corresponding descriptor vectors $\boldsymbol{x}^{(t)}$ have the highest *mutual information* [3]

$$I(\boldsymbol{y}, \boldsymbol{x}^{(t)}) = \Pr(\boldsymbol{y}, \boldsymbol{x}^{(t)}) \log \frac{\Pr(\boldsymbol{y}, \boldsymbol{x}^{(t)})}{\Pr(\boldsymbol{y}) \Pr(\boldsymbol{x}^{(t)})} \qquad (8)$$

with the class labels $\boldsymbol{y}$. Equivalently (8) can be expressed as

$$I(\boldsymbol{y}, \boldsymbol{x}^{(t)}) = H(\boldsymbol{y}) - H(\boldsymbol{y} \mid \boldsymbol{x}^{(t)}) \qquad (9)$$

where $H(\boldsymbol{v}) = -\Pr(\boldsymbol{v}) * \log(\Pr(\boldsymbol{v}))$ is the *Shannon entropy*. Note that the mutual information measure can be viewed as a nonlinear correlation coefficient. The first drawback of this method is that in order to calculate the probabilities in (8) one has to approximate the corresponding probability densities from the data, which can lead to inaccuracies. The second drawback is that the mutual information measure only considers individual correlations of descriptors with the output, but not of sets of descriptors.

For the later experiments we used a slightly improved version of (8), which is given as

$$\frac{2 * I(\boldsymbol{y}, \boldsymbol{x}^{(t)})}{H(\boldsymbol{y}) + H(\boldsymbol{x}^{(t)})} \qquad (10)$$

This symmetrization gives us a score in [0, 1].

## 4 **Our Method – the** IRRM algorithm

RFE is a powerful and fast descriptor selection algorithm, but as it uses a greedy strategy to perform backward elimination it can lead to suboptimal solutions. In our algorithm (see also

[9]) we wish to combine the speed of RFE as a descriptor ranking algorithm with a method to further improve accuracy of the classifier. Our basic idea is as follows: Given some ranking of all descriptors, we can divide our descriptors in a set $S$ of $m$ descriptors which are used for our classifier and a set $R$ of $d - m$ descriptors which are the removed descriptors. However, there might be descriptors in $R$ which should be combined with some of $S$ to further improve our accuracy, i.e. reduce our regularized risk. If we view our set $R$ as a queue, then naturally the first $\eta$ descriptors are those which should be tested first to improve our performance. Hence we add them to our set $S$. Afterwards we remove the worst $\eta$ descriptors from $S$ according to the RFE criterion (7). These descriptors are then put at the end of the queue (see figure below).

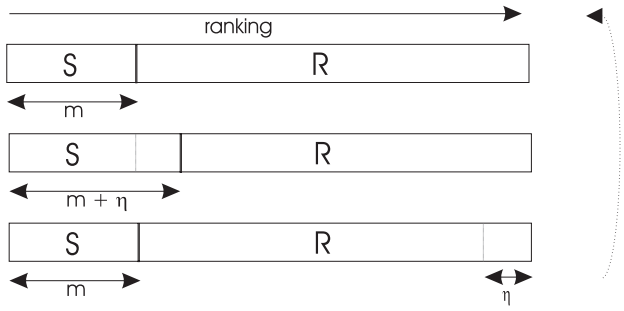


**Figure 1**: Basic idea of the IRRM algorithm

For each descriptor subset $S$ we calculate the regularized risk. If our regularized risk did not improve significantly any more (e.g. less than $10^{-5}$ in 5 successive iterations), we assume the algorithm to be converged. This is usually achieved after a few loops. We then resort the queue by performing RFE and restart the whole algorithm. If again we converge to the same solution, we stop, otherwise we restart the algorithm.

The reason, why we do not resort the queue after each step is, that changing just a few descriptors from the queue will not change our ranking significantly. Hence we would put almost the same descriptors at the beginning of our queue as those which we have removed

before. Additionally, note that a resorting after each step, i.e. running RFE on the complete set $R$, would impose an unacceptably high computational burden.

The details of the algorithm, which we call **Incremental Regularized Risk Minimization (IRRM),** are given below:

**Algorithm IRRM**

perform RFE
$S$ = set of selected features
$R$ = set of removed features
$t \leftarrow 0$
repeat
  $S^{\text{old}} \leftarrow S$
  repeat
    $R_{\text{reg}}^{\text{old}} \leftarrow R_{\text{reg}}$
    compute $R_{\text{reg}}$ for features in $S$
    if $R_{\text{reg}}^{\text{old}} < R_{\text{reg}}$
      restore old $S$
    end
    $C = \eta$ highest ranked features from $R$
    $S \leftarrow S \cup C$
    $R \leftarrow R - C$
    remove $\eta$ features from $S$ according to RFE criterion
    put removed features at end of queue $R$
  until convergence
  resort queue $R$ by means of RFE
  $t \leftarrow t + 1$
until $S == S^{\text{old}}$ AND $t > 1$
return best solution $S^{*}$

We empirically tested η-values of $m$, $m/2$, $0.1m$ and 1 and found η = 1 to perform best. Thus the following results refer to this situation.

# 5 Experiments

## *5.1 Data Set and Preparation*

We used a collected set of Human Intestinal Absorption (HIA) values with 164 structures from the literature, which was also published as a feature selection benchmark data set by Wegner/Fröhlich/Zell [28]. The data set is a collection of Wessel et al. [30] (82 structures), Gohlke/Kissel [10] (49 structures), Palm et al. [20] (8 structures), Balon et al. [1] (11 structures), Kansy et al. [16] (6 structures), Yazdanian et al. [35] (6 structures) and Yee [36] (2 structures).

As descriptors different atom property based descriptors from the JOELib open source library [14] as described in Wegner et al. [27, 29] were used: atom mass (tabulated), valence (calculated, graph connectivity), conjugated environment (calculated, SMARTS based), van der Waals volume (tabulated), electron affinity (tabulated), electronegativity (tabulated, Pauling), graph potentials (calculated, graph theoretical), Gasteiger-Marsili partial charges (calculated, iterative), intrinsic state (calculated), electrotopological state (calculated), electrogeometrical state (calculated), conjugated topological distance (calculated, graph theoretical), conjugated electro toplological state (calculated, graph theoretical). Furthermore we calculated the descriptors set available in MOE [18].

Because all kind of descriptors depend on expert systems to assign the aromaticity, the implicite valence, the pH value correction and the atom type (discrete atom property), the results can only be reproducible, if other authors use exactly the same expert systems and tabulated and calculated values for the atom properties. Thus in general transparency of the system (as it is ensured by MOE and JOELib) is an important issue.

The data set was tested for duplicate molecules, descriptors with missing values were removed, and all descriptors were normalized to mean 0 and standard deviation 1. Additionally all calculated descriptors values were part of this benchmark data set published in our feature selection review paper [28].

For the following SVM model trainings we chose a RBF kernel of width $\sigma = 256$ and soft margin parameter $C = 110$.

### *5.2 Conduction and Results*

In a first experiment we compare the model quality obtained by different descriptor selection algorithms. This gives us an indicator which descriptor algorithm we can trust most. We compare our IRRM algorithm to RFE and mutual information based descriptor selection using 9-fold cross-validation[1]. The $\ell_2$-AROM algorithm was not taken into consideration due to the nonlinearity of the data. The following table shows the results:

**Table 1:** Comparison of the model quality obtained by different descriptor selection algorithms on the HIA data set depending of the number of descriptors to be selected. Results are measured by 9-fold cross-validation. The table contains the average error rates ± standard error (%). The standard SVM with all descriptors achieved 15.76% ± 3.27%.

| # descriptors | IRRM | RFE | mutual information |
|---|---|---|---|
| 10 | **26.67 ± 4.17** | 26.67 ± 4.56 | 35.15 ± 5.55 |
| 20 | **18.75 ± 3.99** | 21.18 ± 4.49 | 32.07 ± 5.76 |
| 30 | **16.99 ± 2.38** | 18.19 ± 2.89 | 26.54 ± 6.28 |
| 40 | **17.61 ± 2.27** | 17.61 ± 2.45 | 26.58 ± 6.43 |
| **50** | **15.76 ± 2.54** | 17.61 ± 2.45 | 23.57 ± 6.63 |
| 100 | **16.99 ± 2.38** | 17.61 ± 2.62 | 19.43 ± 3.21 |
| 250 | **16.37 ± 2.79** | 16.99 ± 2.87 | 17.61 ± 2.93 |

[1] We used 9 folds in order to have a division rest as small as possible when dividing the data into *k* pieces and hence avoiding large deviations of the cross-validation results on each fold.

As one can see our IRRM algorithm gives always better results than RFE and mutual information. Our algorithm was the only one that achieved the same generalization performance as the standard SVM using all descriptors. In this case IRRM selected only 50 out of 2929 descriptors.

Next we trained IRRM on all data to see which are the 50 most relevant descriptors. They are given in table 2. The high relevance of the TPSA and the PEOE_VSA_POL descriptor is consistent with the literature [29, 30].

By using only these descriptors and training a SVM with 9-fold cross-validation we obtained 12.67% ± 2.74% error rate. However, this result should be taken with care, because IRRM was already trained on *all* data before. Hence the selected descriptors are adapted to be well suited for the whole data set a priori.

## 6 Using the IRRM Algorithm for Regression

Similar to the problem of finding an appropriate set of descriptors for a classification problem, we can ask ourselves how we can find a good subset of descriptors for a regression problem, e.g. for predicting the aqueous solubility.

Basically we can view this problem in the same way as in section 2. The only difference is that now we measure errors by means of the so called ε-insensitive loss function [25]

$$\ell_\varepsilon(y, f(\boldsymbol{x})) = \begin{cases} 0 & |y - f(\boldsymbol{x})| \le \varepsilon \\ |y - f(\boldsymbol{x})| - \varepsilon & \text{otherwise} \end{cases}$$

That means we only count errors which are absolutely bigger than some small tolerance ε. Using this loss function in the regularized risk (4) will bring us to Support Vector Regression (SVR) [25]. Like in SVMs for classification, in SVR $\|\boldsymbol{w}\|$ is a measure for the models' complexity. In SVR it can be interpreted as a measure for the *flatness* of the regression hyperplane in feature space. Hence, in order to perform descriptor selection, we seek the

combination of descriptors which keeps the flatness of our regression function small. Thus in practice we only have to change the used loss function in our IRRM algorithm to obtain a descriptor selection algorithm for regression. Similar considerations can be made for the RFE algorithm. The mutual information measure is applicable to regression anyway.

# 7 Experiments

## *7.1 Data Set and Preparation*

For predicting the aqueous solubility (logS) we used the data set of Huuskonen [13], which was also published as a benchmark data set including also a MOE [18] and JOELib [14] descriptor set. The data set was tested for duplicate molecules, descriptors with missing values were removed, and all descriptors (including the aqueous solubility) were normalized to mean 0 and standard deviation 1. The normalized logS values this way have a range from –4.36 to 2.11. It must be mentioned that previous publications with and without using descriptor selection algorithms [27] use no cross-validation to avoid overfitting/underfitting of the hyptheses, and hence the generalization ability of these experiments must be seen critically (see also [28]).

For the following SVR model trainings we chose a RBF kernel of width $\sigma = 68$ and soft margin parameter $C = 10$. The error tolerance $\varepsilon$ was set to 0.12.

## *7.2 Conduction and Results*

Again we first compare the model quality obtained by the different descriptor selection algorithms. We used 8-fold cross-validation. Table 2 shows the mean squared *test* error rates and, for the sake of completeness, also the mean squared *training* errors in brackets.

We also include the mean squared correlations $r^2 = \frac{1}{l}\sum_{l=1}^{k}\frac{\mathrm{cov}(\boldsymbol{p}_l, \boldsymbol{t}_l)}{\sigma_{\boldsymbol{p}_l} \cdot \sigma_{\boldsymbol{t}_l}}$ between model predictions $\boldsymbol{p}$ and true values $\boldsymbol{t}$ in table 3. The index $l$ refers to the $l$th of $k$ folds.

**Table 2:** Comparison of the model quality obtained by different descriptor selection algorithms on the Huuskonen data set depending on the number of descriptors to be selected. Results are measured by 8-fold cross-validation. The table contains the mean squared *test* error rates ± standard error and the mean squared *training* errors in brackets. The standard SVR with all descriptors achieved mean squared *test* error of 0.101 ± 0.008 and a mean squared *training* error of $0.03 \pm 0.3 \cdot 10^{-3}$. The normalized logS values have a range from –4.36 to 2.11.

| # descriptors | IRRM | RFE | mutual information |
|---|---|---|---|
| 10 | **0.161 ± 0.007** ($0.139 \pm 1.2 \cdot 10^{-3}$) | 0.162 ± 0.007 ($0.139 \pm 1.4 \cdot 10^{-3}$) | 0.617 ± 0.017 ($0.569 \pm 67.5 \cdot 10^{-3}$) |
| 20 | **0.142 ± 0.006** ($0.123 \pm 2.6 \cdot 10^{-3}$) | 0.145 ± 0.007 ($0.133 \pm 2.3 \cdot 10^{-3}$) | 0.526 ± 0.018 ($0.274 \pm 109.6 \cdot 10^{-3}$) |
| 50 | **0.121 ± 0.006** ($0.11 \pm 1 \cdot 10^{-3}$) | 0.124 ± 0.006 ($0.117 \pm 1.6 \cdot 10^{-3}$) | 0.3 ± 0.01 ($0.196 \pm 44.5 \cdot 10^{-3}$) |
| 100 | **0.115 ± 0.007** ($0.103 \pm 1.1 \cdot 10^{-3}$) | **0.115 ± 0.006** ($0.105 \pm 0.8 \cdot 10^{-3}$) | 0.18 ± 0.007 ($0.128 \pm 4.3 \cdot 10^{-3}$) |
| 250 | **0.103 ± 0.006** ($0.084 \pm 0.9 \cdot 10^{-3}$) | **0.103 ± 0.006** ($0.085 \pm 0.6 \cdot 10^{-3}$) | 0.16 ± 0.007 ($0.107 \pm 0.8 \cdot 10^{-3}$) |

**Table 3:** Comparison of the model quality obtained by different descriptor selection algorithms on the Huuskonen data set depending on the number of descriptors to be selected.

The table contains the mean squared correlations (%) ± standard error (%) for 8-fold cross-validation. The standard SVR with all descriptors achieved 90.25% ± 0.08%.

| # descriptors | IRRM | RFE | mutual information |
|---|---|---|---|
| 10 | **86.12 ± 1.13** | 85.96 ± 1.08 | 39.56 ± 1.94 |
| 20 | **86.9 ± 1.04** | 85.87 ± 1.02 | 48.08 ± 2.52 |
| 50 | **88.53 ± 0.79** | 87.61 ± 0.83 | 64.63 ± 2.58 |
| 100 | **88.91 ± 0.74** | 88.43 ± 0.7 | 81.98 ± 1.27 |
| 250 | **89.71 ± 0.77** | 89.66 ± 0.76 | 83.97 ± 1.2 |

In his original paper Huuskonen [13] used a Neural Network to predict aqueous solubility. Using a training set of 884 and a test set of 413 examples he achieved a squared correlation of 92% ± 1.6% on the test set. Once again note that this result is much less reliable than ours, since Huuskonen did not perform cross-validation. Hence one should be very careful comparing our results with his.

Our IRRM algorithm in all cases achieved slightly (though not significantly) better cross-validation results than RFE. With just 50 descriptors out of 2808 one can obtain a rather good model, and with 250 descriptors IRRM can achieve a performance which is almost the same as when using all descriptors. Note that one might gain a little more by adding more descriptors, but we restricted ourselves to a maximum of 250 here in order to have a good trade-off between the number of descriptors (and hence training time) and model prediction quality.

Like in the classification case we trained IRRM one the whole data set to extract the 50 most relevant descriptors. They are given in table 5.

By using only the 50 most relevant descriptors and training a SVR with 8-fold cross-validation we achieved a mean quadratic error of 0.114 ± 0.005 ($r^2$ = 88.67% ± 0.7%). By using the 250 best descriptors we achieved a mean quadratic error of 0.100 ± 0.006 ($r^2$ =

89.96% ± 0.76%). As stated in the previous section these last two results should be taken with care, since IRRM was trained on the whole data set before.

## 8 Conclusion

We have presented a new technique for descriptor subset selection with Support Vector Machines, which works by incremental regularized risk minimization and is hence well founded on insights of modern Machine Learning theory. Our IRRM algorithm is efficient and achieves a better model quality than other state-of-the-art algorithms on a HIA data set. As it views the problem of selecting good descriptors in a Machine Learning context, it aims at selecting descriptor subsets rather than concentrating on individual descriptors independently. The descriptors found by our method are consistent with the literature. We like to point out again that our method is fully deterministic, which makes it easier to reproduce results and to compare it to solutions found by other researchers. In contrast to many other previous works we carefully evaluated the quality of our model by using cross-validation. This is a step, which from our point of view should always be done, even if one is interested only in building a single final model.

We generalized our descriptor selection technique to deal with regression problems, such as the Huuskonen data set, by means of Support Vector Regression. Here we found a good model with less than 2% of the original descriptors, and with 250 descriptors one can achieve a model quality which is almost the same as when using all descriptors.

We hope that this work will attract other researchers' interest in modern methods from Machine Learning and the problem of selecting good molecular descriptors, and we hope that this is a positive contribution for building better and more reliable QSAR models in the future.

## Appendix

**Table 4:** Descriptors for the HIA dataset selected by IRRM

| HIA descriptors (alphabetical order) |
|---|
| Burden_modified_eigenvalues:Atom_in_conjugated_environment:0 |
| Burden_modified_eigenvalues:Atom_in_conjugated_environment:1 |
| Burden_modified_eigenvalues:Atom_mass:3 |
| Burden_modified_eigenvalues:Electrogeometrical_state_index:3 |
| Burden_modified_eigenvalues:Electron_affinity:3 |
| Fraction_of_rotatable_bonds |
| hydrogen_donors |
| Number_of_acidic_groups |
| PEOE_VSA_POL |
| PEOE_VSA_PPOS |
| PEOE_VSA+3 |
| PEOE_VSA+4 |
| PEOE_VSA-5 |
| PolarSurfaceArea |
| RDF_B100.0:Atom_in_conjugated_environment:14 |
| RDF_B100.0:Atom_in_conjugated_environment:35 |
| RDF_B100.0:Atom_in_conjugated_environment:41 |
| RDF_B100.0:Atom_in_conjugated_environment:5 |
| RDF_B100.0:Electrogeometrical_state_index:45 |
| RDF_B100.0:Electrotopological_state_index:13 |
| RDF_B100.0:Electrotopological_state_index:45 |
| RDF_B100.0:Gasteiger_Marsili:12 |
| RDF_B100.0:Gasteiger_Marsili:20 |
| RDF_B100.0:Intrinsic_state:45 |
| RDF_B200.0:Atom_in_conjugated_environment:14 |

| |
|---|
| RDF_B200.0:Atom_in_conjugated_environment:35 |
| RDF_B200.0:Atom_in_conjugated_environment:41 |
| RDF_B200.0:Atom_in_conjugated_environment:5 |
| RDF_B200.0:Atom_mass:34 |
| RDF_B200.0:Electrotopological_state_index:13 |
| RDF_B200.0:Electrotopological_state_index:45 |
| RDF_B200.0:Gasteiger_Marsili:28 |
| RDF_B200.0:Gasteiger_Marsili:47 |
| RDF_B200.0:Intrinsic_state:34 |
| RDF_B200.0:Intrinsic_state:45 |
| RDF_B25.0:Atom_in_conjugated_environment:3 |
| RDF_B25.0:Atom_in_conjugated_environment:4 |
| RDF_B25.0:Atom_in_conjugated_environment:41 |
| RDF_B25.0:Conjugated_electrotopological_state_index:46 |
| RDF_B25.0:Intrinsic_state:41 |
| RDF_B5.0:Atom_in_conjugated_environment:40 |
| RDF_B5.0:Atom_in_conjugated_environment:41 |
| RDF_B5.0:Gasteiger_Marsili:45 |
| RDF_B5.0:Gasteiger_Marsili:46 |
| SlogP |
| SlogP_VSA0 |
| SlogP_VSA2 |
| SMR_VSA4 |
| TPSA |
| vsa_don |

**Table 5:** 50 most relevant descriptors for the Huuskonen dataset selected by IRRM

| **Huuskonen descriptors (alphabetical order)** |
|---|
| a_acc |
| a_ICM |
| a_nS |
| Auto_correlation:Intrinsic_state:1 |
| Burden_modified_eigenvalues:Atom_van_der_waals_volume:1 |
| Burden_modified_eigenvalues:Electrotopological_state_index:0 |
| Burden_modified_eigenvalues:Electrotopological_state_index:1 |
| chi0_C |
| chi0v_C |
| chi1v_C |
| glob |
| logP(o/w) |
| PEOE_RPC- |
| PEOE_RPC+ |
| PEOE_VSA_NEG |
| PEOE_VSA-0 |
| pmi |
| pmiX |
| RDF_B100.0:Atom_valence:13 |
| RDF_B100.0:Atom_valence:6 |
| RDF_B100.0:Atom_van_der_waals_volume:7 |
| RDF_B100.0:Electron_affinity:7 |
| RDF_B100.0:Electronegativity_pauling:7 |
| RDF_B100.0:Electrotopological_state_index:7 |
| RDF_B200.0:Atom_valence:6 |
| RDF_B200.0:Atom_van_der_waals_volume:7 |
| RDF_B200.0:Electron_affinity:7 |

| |
|---|
| RDF_B200.0:Electronegativity_pauling:26 |
| RDF_B200.0:Gasteiger_Marsili:6 |
| RDF_B200.0:Intrinsic_state:7 |
| RDF_B25.0:Atom_mass:10 |
| RDF_B25.0:Electron_affinity:18 |
| RDF_B25.0:Electron_affinity:5 |
| RDF_B25.0:Electron_affinity:7 |
| RDF_B5.0:Atom_mass:9 |
| RDF_B5.0:Gasteiger_Marsili:16 |
| RDF_B5.0:Intrinsic_state:4 |
| RDF_B5.0:Intrinsic_state:5 |
| RDF_B5.0:Intrinsic_state:6 |
| RDF_B5.0:Intrinsic_state:7 |
| SlogP |
| SlogP_VSA0 |
| SlogP_VSA8 |
| SMR_VSA2 |
| std_dim3 |
| TPSA |
| VAdjEq |
| vsa_hyd |
| vsa_other |
| Weight |

## References

[1] K. Balon, B. Riebesehl, B. Müller, Drug Liposome Partitioning as a Tool for the Prediction of Human Passive Intestinal Absorption, *Pharm. Res.,* **1999**, *16*, 882-

888.

[2] A. L. Blum, P. Langley, Selection of Relevant Features and Examples in Machine Learning, *Artificial Intelligence*, **1997**, *97(12),* 245 – 271.

[3] B. Bonnlander, A. Weigend, Selecting input variables using mutual information and nonparametric density estimation, in: *Proc. 1994 Int. Symp. on Artificial Neural Networks*, **1994**, pp. 42 - 50.

[4] E. Byvatov, U. Fechner, J. Sadowski, G. Schneider, Comparison of Support Vector Machine and Artificial Neural Network Systems for Drug/Nondrug Classification, *J. Chem. Inf. Comput. Sci.*, **2003**, *43(6),* 1882 - 1889.

[5] E. Byvatov, G. Schneider, *J. Chem. Inf. Comput. Sci.*, **2004**, ASAP Article, DOI 10.1021/ci0342876.

[6] C. Cortes, V. Vapnik, Support vector networks, *Machine Learning*, **1995**, *20*, 273 – 297.

[7] S. Davies, S. Russel, NP-Completeness of Searches for Smallest Possible Feature Sets, in: *Proc. of the 1994 AAAI Fall Symposion on Relevance*, **1994**, pp. 37 – 39.

[8] H. Fröhlich, O. Chapelle, B. Schölkopf, Feature Selection for Support Vector Machines by Means of Genetic Algorithms, in: *Proc. 15th IEEE Int. Conf. on Tools with AI*, **2003**, pp.142 - 148.

[9] H. Fröhlich, A. Zell, Feature Subset Selection for Support Vector Machines by Incremental Regularized Risk Minimization, *IEEE Int. Joint Conf. on Neural Networks (IJCNN)*, **2004**, accepted paper.

[10] H. Gohlke, F. Dullweber, W. Kamm, J. März, T. Kissel, G. Klebe, Prediction of Human Intestinal Absorption using a combined 'Simulated Annealing/Backpropagation Neural Network' Approach. *Rational Approaches Drug Des.,* **2001***,* 261-270.

[11] I. Guyon , A. Elisseeff, An Introduction into Variable and Feature Selection, *Journal of Machine Learning Research Special Issue on Variable and Feature Selection*, **2003**, *3*, 1157 - 1182.

[12] I. Guyon, J. Weston, S. Barnhill, V. Vapnik, Gene Selection for Cancer Classification using Support Vector Machines, *Machine Learning*, **2002**, *46*, 389 – 422.

[13] J. Huuskonen, Estimation of Aqueous Solubility for Diverse Set of Organic Compounds Based on Molecular Topology, *J. Chem. Inf. Comput. Sci.*, **2000**, *40*, 773 - 777.

[14] JOELib, http://joelib.sourceforge.net/.

[15] G. John, R. Kohavi, K. Pfleger, Irrelevant features and the subset selection problem, in: *Machine Learning: Proc. of the 11th Intern. Conf.*, **1994**, pp.121 - 129.

[16] M. Kansy, F. Senner, Gubernator, K. Physicochemical High Throughput Screening: Parallel Artificial Membrane Permeation Assay in the Description of Passive Absorption Processes, *J. Med. Chem.,* **1998***, 41*, 1007-1010.

[17] R. Kohavi, G. John, Wrappers for Feature Subset Selection, *Artificial Intelligence*, **1997**, *97(12)*, 273 - 324.

[18] MOE (Molecular Operating Environment), Chemical Computing Group Inc., **2003**.

[19] N. Nikolova, J. Jaworska, Approaches to Measure Chemical Similarity - a Review, *QSAR & Combinatorial Science*, **2003**, *22*, 9-10.

[20] K. Palm, P. Stenburg, K. Luthman, P. Artursson, Polar Molecular Surface Properties Predict the Intestinal Absorption of Drugs in Humans. *Pharma. Res.,* **1997**, *14*, 568-571.

[21] A. Rakotomamonjy,Variable Selection Using SVM based Criteria, *Journal of Machine Learning Research Special Issue on Variable and Feature Selection*, **2003**, *3*, 1357 - 1370.

[22] S. Salcedo-Sanz, M. Prado-Cumplido, F. Perez-Cruz, C. Bousono-Calzon, Feature Selection via Genetic Optimization, in: *Proc. Int. Conf. On Artificial Neural Networks (ICANN)*, **2002**, pp. 547 - 552.

[23] B. Schölkopf, C. Burges, V. Vapnik, Extracting support data for a given task, in: U. N. Fayyad and R. Uthurusamy (Ed.), *First Int. Conf. for Knowledge Discovery and Data Mining*, Menlo Park, **1995**, AAAI Press.

[24] H. Vafaie, K. De Jong, Evolutionary feature space transformation, in: H. Liu, H. Motoda (Ed.), *Feature Extraction, Construction and Selection: a data mining perspective*, Kluwer, **1998**, pp. 307 - 323.

[25] V. Vapnik, *The Nature of Statistical Learning Theory*, Springer, New York, NY, **1995**.

[26] V. Vapnik, *Statistical Learning Theory*, John Wiley and Sons, New York, **1998**.

[27] J. Wegner, A. Zell, Prediction of Aqueous Solubility and Partition Coefficient Optimized by a Genetic Algorithm Based Descriptor Selection Method, *J. Chem. Inf. Comput. Sci.*, **2003**, *43*, 1077-1084.

[28] J. Wegner, H. Fröhlich, A. Zell, Feature Selection for Descriptor based Classificiation Models. 1. Theory and GA-SEC Algorithm, *J. Chem. Inf. Comput. Sci.*, **2003**, *44*.

[29] J. Wegner, H. Fröhlich, A. Zell, Feature selection for Descriptor based Classification Models. 2. Human Intestinal Absorption (HIA), *J. Chem. Inf. Comput. Sci.*, **2003**, *44*.

[30] M. D. Wessel, P. C. Jurs, J. W. Tolan, S. M. Muskal, Prediction of Human

Intestinal Absorption of Drug Compounds from Molecular Structure, *J. Chem. Inf. Comput. Sci.*, **1998**, *38*, 726 - 735.

[31] J. Weston, A. Elisseeff, B. Schölkopf, M. Tipping, Use of the zero-norm with linear models and kernel methods, *JMLR special Issue on Variable and Feature Selection*, **2002**, *3*, 1439 - 1461.

[32] J. Weston, S. Mukherjee, O. Chapelle, M. Pontil, T. Poggio, V. Vapnik, Feature selection for SVMs, in: S. Solla, T. Leen, K.-R. Müller (Ed.), *Adv. in Neural Inf. Proc. Syst. 13*, **2001**, MIT Press.

[33] D. Wolpert , W. Macready, No Free Lunch Theorems for Search, Technical Report SFI-TR-95-02-010, Santa Fee Institute, **1995**.

[34] D. Wolpert , W. Macready, No Free Lunch Theorems for Optimization, in: *Proc. IEEE Transactions on Evolutionary Computation*, **1997**, *1*, pp. 67 - 82.

[35] M. Yazdanian, S. Glynn, J. Wright, A. Hawi, Correlating Partitioning and Caco-2 Cell Permeability of Structurally Diverse Small Molecular Weight Compounds, *Pharm. Res.*, **1998**, *15*, 1490-1494.

[36] S. Yee, In Vitro Permeability Across Caco-2 Cells (Colonic) Can Predict In Vivo (Small Intestinal) Absorption in Man-Fact or Myth, *Pharm. Res*., **1997**, *14*, 763-766.