Support Vector Machine Regression in Chemometrics

Ayhan Demiriz *
E-Business Department, Verizon Inc.,
919 Hidden Ridge, Irving, TX 75038
E-mail:ayhan.demiriz@verizon.com

Kristin P. Bennett
Dept. of Mathematical Sciences
Rensselaer Polytechnic Institute
Troy, NY 12180
E-mail:bennek@rpi.edu

Curt M. Breneman
Dept. of Chemistry
Rensselaer Polytechnic Institute
Troy, NY 12180
E-mail:brenec@rpi.edu

Mark J. Embrechts
Dept. of Decision Sci. and Eng. Systems
Rensselaer Polytechnic Institute
Troy, NY 12180
E-mail:embrem@rpi.edu

Abstract

Predicting the biological activity of a compound from its chemical structure is a fundamental problem in drug design. The ability exists to generate vast amounts of potential pharmecutical compounds. Statistical and machine learning methods can provide an efficient means of estimating the bioreponses of these compounds in order to expedite drug design. In this paper we develop a Support Vector Machine Regression (SVMr) methodology for estimating the bioresponse of molecules based on the large sets of descriptors. Since the concerned data is characterized by large numbers of descriptors and very few data points, we adapt SVMr model selection and bagging strategies in order to avoid overfitting. The proposed approach compares very favorably with Partial Least Squares (PLS), a well-known and commonly used method in chemometrics, on the performance of Quantitative Structure-Activity Relationships (QSAR) analysis based on real chemistry data.

*This work has been partially done while the first author was a graduate student in Dept. of Decision Sciences and Engineering Systems at Rensselaer Polytechnic Institute. Project url: *http://www.drugmining.com.*

Keywords: Partial Least Squares, Support Vector Machine Regression, Chemometrics, Quantitative Structure-Activity Relationships.

1 Introduction

The demand for the drugs to fight antibiotic-resistant bacterial infections, cancer, AIDS and other common diseases is ever increasing. International drug companies also face accusations from developing and third world countries for selling their drugs at very inflated prices [10]. Because of both the complexity of diseases that humans face and the high costs of the drugs to fight these diseases, scientists from various disciplines feel the pressure to design and develop efficient drugs in a very short time with minimal cost. The conventional drug design process is very long and expensive. Finding promising compounds in the early phase of the drug design process greatly reduces the time and the cost of such processes. One goal of the drug design process is to find a relatively small compound (ligand) that consists of a few dozens of atoms that binds with a receptor cavity of certain proteins or enzymes. Receptor-ligand bindings can create necessary biological activity that leads to the desired pharmacological features.

Developments in automated compound generation enables the creation of vast compound (molecule) databases. Two effective ways of approaching such databases exist: Rational Design and Combinatorial Synthesis. Combinatorial synthesis in which many different ligands are synthesized in parallel and tested via a high throughput screening is beyond the scope of this paper. The basic idea behind rational design is to analyze the structure/activity correlation in molecule databases in order to predict biological activities (and possible chemical reactivity and toxicity) of molecules of interest. Determining the relationship between the molecule structure and its biological activity with certain molecules enables one to analyze the behaviour of other similar molecules.

Quantitative Structure-Activity Relationships (QSAR) analysis plays an essential role in the rational drug design process. QSAR analysis can be summarized as the task of discovering new molecules with desired pharmaceutical properties, especially in the early phase of the drug design process, from a pool of molecules by analyzing structure/activity relationships. The main underlying assumption in QSAR analysis is that the biological activities of a group of compounds can be explained by analyzing their respective structural (physical) and electronic features [4]. Successful QSAR analysis will expedite drug design creating enormous economic and health benefits for both public and private sectors. On the other hand, mistakes in QSAR analysis may cause the undesired costs of pursuing some improper compounds in the drug design process and/or discarding potentially good compounds from the drug design process.

Various analytical tools from statistics and machine learning are used in QSAR analysis including predictive modeling (classification and regression), visualization, exploratory data analysis through principal components and cluster analysis. In addition to standard tools used in other fields, specialized tools have been developed by chemometricians based on heuristic reasoning and intuitive ideas [6]. The most popular regression tool among the chemometricians is Partial Least Squares (PLS), which is little known and used in other fields. The popularity of PLS comes from its ability to model very high dimensional data with very few observations on hand. In most QSAR (chemometrics) applications the ratio of the number of predictors to the number of observations is very high. The details of PLS are given in Section

3.

Support Vector Machines (SVM) have recently emerged as an alternative regression tool [11]. The strength of SVM regression (SVMr) comes from its ability to represent very high dimensional input space (predictors) through kernel functions with great resistance to overfitting. The resulting SVM model is independent from the dimensionality of the input space. As discussed in Section 4, instead of using $n \times p$ data matrix, where n and p are the number of observations and the number of predictors (features) respectively, an $n \times n$ kernel matrix is used in the SVM model. Since QSAR applications require the analysis of data with very few observations but a very high number of dimensions, SVM regression appears to be a suitable analysis tool for chemometrics. For a successful SVM model, care must be taken in the selection or parameters in the objective and kernel function. We examine how to perform this model selection task when very little data is available. Multiple validation sets are used within a grid search to construct a set of candidate models. The resulting models are bagged or averaged in order to reduce variance. We use the linear program representation proposed in [11] in order to exploit efficient optimization algorithms available for linear programs. Since the size of the kernel matrix ($n \times n$) is very small in QSAR data, we can extensively search the parameter space with the help of efficient reoptimization without solving each the underlying optimization problem from scratch. The details of our SVM regression implementation are given in Section 5.

In Section 6, we give results of a comparison of SVMr and PLS on seven QSAR datasets. Experimental results indicate that SVM regression outperforms PLS in QSAR analysis. We then conclude our paper in Section 7.

2 Related Work

This paper compares only two approaches used in QSAR analysis, PLS and SVMr. PLS is commonly used in chemometrics as an industrial standard. Other solutions to the drug design process exist in the literature. Due to space limitations, most of them can not be addressed in this paper. Nevertheless, we will mention a few related works in this section. In our work we focus on QSAR as a regression problem, but predicting biological activities can also be posed as a classification problem to determine whether or not a molecule is worth further consideration in terms of biological activity. In [4], SVM classifier was compared with Radial Basis Functions (RBF) network, neural networks, decision trees and nearest neighbor classifier. SVM with RBF kernel was the best among the investigated methods as measured 5-fold cross-validation (CV).

The pose of the ligand determines some of the structural features in QSAR data. This problem also resembles that of hand-writing recognition. In [5], a neural network is trained to find the best posing of the compound that is used in predicting the biological activity. So called dynamic reposing is compared with other posing methods known in hand-writing recognition such as standard posing, tangent-propagation and tangent distance [5]. Again the problem in [5] is to predict the biological activity of compounds depending on their poses. Dynamic reposing is an Expectation-Maximization (EM) method in a sense where the best pose is found first and then depending on the posing biological activity is predicted. Process continues iteratively until it reaches convergence (no change in poses). Dynamic reposing is proposed to avoid the conformation search. But efficient algorithms exist to search conformations which have low energy [9].

Machine learning algorithms are also used to select existing and find new features (descriptors) in structure/activity prediction. Inductive Logic Programming (ILP) is used in [12] to generate new features based on available structural properties. It is indicated in [12] that new features based on ILP enhances the predictability of the biological activity using linear regression. The variable selection problem was addressed from the QSAR analysis point of view in [8]. Neural network sensitivity analysis and neural bootstrapping were coupled in [8] to analyze QSAR data using a successful variable selection approach. Chemometricians most commonly used the PLS method. In the next section we briefly explain the PLS method and its usage.

3 Partial Least Squares

Since the introduction of PLS to chemometrics in mid-70's (it was originally proposed by H. Wold, an econometrician, in the 60's), it has been used widely as an alternative method to Ordinary Least Squares (OLS) regression. The driving force behind this move was the inability of OLS to handle problems with high collinearity among the predictors and very few observations. For many years statisticians did not pay noticable attention to PLS until the early 90's [6]. Statistically PLS, Principal Component Regression (PCR) and ridge regression have many similarities [6]. A through comparison of PLS, PCR and ridge regression can be found in [6] for interested readers. But all the three methods try to shrink the solution vector from the OLS solution in directions where predictors have higher variations.

PLS is an iterative algorithm. PLS' robustness comes from the fact that at each iteration PLS shrinks (projects) the OLS solution towards the maximum correlation between the residual error (of response y) and the input data ($\mathbf{x}$). PLS recursively computes the orthogonal projections of the input data and performs single variable regressions along these projections on the residual error of the previous iteration [14]. Thus the regression solution depends on the decomposition of both response and predictor variables simultaneously.

PLS' iterative structure allows it to find a new factor at each iteration. PLS avoids the collinearity problem by projecting high dimensional observed variables into lower dimensional factors. The degree of bias is dependent on the choice of the number of factors. The smaller the number of factors, the larger bias. But a large number of factors will produce high variance. Usually cross validation is used to determine the number of factors used in the solution. These factors are linear in the input data $\mathbf{x}$. In the case of non-linear relationship between response and predictors, PLS might end up with a solution that has very large bias. Like all other linear methods, this is a weakness of PLS. Nevertheless, PLS' robustness to overfitting helps keep its prominent position as an analytical tool in chemometrics. In the next section, we introduce SVM regression as a new tool in chemometrics. This is one of the very first attempts to use SVM regression for chemometrics in the literature.

4 SVM Regression

As a new machine learning technique SVM was proposed in the pattern recognition context. For example, SVM was used in [4] to solve a classification problem. The basic properties of classification SVMs also hold for regression, thus successful SVM regression models exist [11]. We briefly introduce the linear programming SVM regression in [11] in this section.

Let's assume we are given a training set of $(x_1, y_1), \ldots, (x_n, y_n)$ where $x_i \in \mathbb{R}^p$. The objective of the learning process (in this case regression) is to find a function f which minimizes the risk function: $R[f] = \int_{\mathbb{R}^p} l(f, x, y) dP(x, y)$, where P is the underlying probability distribution and l is the appropriate loss function e.g. square loss function. Since the true distribution of P is unknown, we need to minimize the *empirical risk*, $R^{\varepsilon}_{emp}[f]$, as measured on the training data. Many recent formulations of SVM regression use the ε-insensitive loss function ($|y - f(x)|_{\varepsilon} = \max\{0, |y - f(x)| - \varepsilon\}$) to minimize the regularized risk:

$$R_{reg} := ModelCapacity + C \cdot R^{\varepsilon}_{emp}[f] \tag{1}$$

where $ModelCapacity$ characterizes the model complexity and C is the constant regularization parameter (trade-off). For ε-insensitive loss function, the empirical risk becomes $R^{\varepsilon}_{emp}[f] := \frac{1}{n} \sum_{i=1}^{n} |y_i - f(x_i)|_{\varepsilon}$.

The power of SVM comes from the kernel representation that allows a non-linear mapping of input space to a higher dimensional feature space. The regression function can be written as a linear combination of mapped training examples x_i. Thus one obtains the well known kernel expansion of function f as:

$$f(x) = \sum_{i=1}^{n} \alpha_i k(x_i, x) + b$$

where α's are the multipliers, $k(x, x')$ is the kernel function and b is the bias.

For the linear program (LP) model, the $ModelCapacity$ in Eq. 1 is measured to produce the function f with the smallest nonnegative combination of the patterns x_i: $R_{reg} := \sum_{i=1}^{n} \alpha_i + C \cdot R^{\varepsilon}_{emp}[f]$. In the ε-insensitive loss function term, ε is typically a user specified parameter. If we also want to include ε into our optimization problem then we have the following LP formulation of SVM regression:

$$\begin{array}{ll} \min\limits_{\alpha, b, \xi, \xi^*, \varepsilon} & \sum_{i=1}^{n} \alpha_i + \dfrac{C}{n} \sum_{i=1}^{n} \xi_i + \xi_i^* + C\nu\varepsilon \\ \text{s.t.} & \sum_{i=1}^{n} \alpha_i k(x_i, x_j) + b - y_j \leq \varepsilon + \xi_j \\ & y_j - \sum_{i=1}^{n} \alpha_i k(x_i, x_j) - b \leq \varepsilon + \xi_j^* \quad j = 1, \ldots, n \\ & \alpha, \xi, \xi^* \geq 0, \\ & \alpha \in \mathbb{R}^n, b \in \mathbb{R}, \xi, \xi^* \in \mathbb{R}^n, \end{array} \tag{2}$$

At optimality $\xi_i + \xi_i^* = |y - f(x)|_{\varepsilon}$.

To successfully apply the SVM regression LP (2), one must prepare the data, select the model parameters (C, ν), select the kernel function and associated parameters and then optimize the models. In this work, we used the radial basis function kernel with parameter σ

$$k(x_i, x_j) = \exp(-||x_i - x_j||^2 / \sigma^2) \tag{3}$$

In the next section we describe our methodology for accomplishing these steps for QSAR problems.

5 Our SVMr Methodology

QSAR is a very difficult inference problem because the dimensionality of the input space is very high, (600 to 1000 variables) and the amount of training data is small (frequently less than 100 observations). The risk of overfitting and obtaining poor generalization is great. Careful model selection and validation is essential for good results using SVMr. We utilize a multiple step process. First the data is prepared by normalizing it and removing high variance variables. Then we perform a model selection procedure by optimizing the model error as measured on validation sets. This produces multiple candidate SVM models. Since the validation sets are of necessity very small, there is a very high variance in the set of SVM models. Thus the final step is to average or bag the models in order to improve the final model accuracy. These steps are described in detail in this section.

Successful data analysis starts with careful data preparation. The first step for both PLS and SVMr is to normalize the data. Often chemical and physical variables have different units of measurements. Thus variables vary in very different ranges. This is very problematic for any data analysis tool. Having different magnitude of variables also affects the kernel function. Certain variables might have much more importance in the learning process than they deserve or vice versa. The SVMr method in this paper is provided standard normalized data. Since the response variables are log-scaled response, they are not normalized. We use the RBF kernel function, the most used kernel in the SVM regression literature. Typically SVM models with RBF kernel functions perform better with normalized data. In addition in order to prevent outliers' effects on the kernel function mapping, we screened our original data to remove the variables that have values outside of the $\pm 4\sigma$ range. No other feature selection is done for SVMr in this paper. But feature selection can be added to further improve the performance of SVMr, PLS, and other learning methods [8].

The next step in the SVMr process is to select the model parameters and optimize the model. As seen from Eq. 2, there are three parameters in this particular LP formulation: C, ν and the parameter σ^2 of the kernel function k. Optimization of the LP can be readily accomplished using commercial LP packages. We use CPLEX 6.5 [7] to solve our LP model. The primary research challenge is how to select the model parameters. Our strategy is to optimize the model error as measured on a validation set across a fixed set of possible parameters. The core of our SVM regression implementation is to use reoptimization to speed up the exhaustive parameter search. CPLEX can very efficiently reoptimize the LP after changes in the objective parameters, C and ν. Thus, we build three loops in our program to change the parameters. In the most inner loop, we change ν and in the next loop we change C. Finally in the outer loop, we change the kernel parameter. The reason behind this is that we put the parameter ν in the most inner loop because we expect that any change in ν will effect the current LP solution the least. Then parameter C and then the parameter of the kernel function k. C and ν only affect the objective function. On the other hand, change in the kernel parameter affects the constraint matrix of the LP. Reoptimization after changes in the constraint matrix is much more costly.

To make our model strategy robust on QSAR problems, we have one final step. Typically we take our validation set to be 10% of the training set. For small QSAR problems in high dimensions, changes in the validation set can result in very different models with high variance in accuracy. To make the method more robust, we construct several models using different validation sets, and then average them to produce a final model. This is a form of bagging [1]. Since a linear combination of SVMr functions is another SVMr function, the final model is still a SVMr function but it is far more robust than any function constructed using a single validation set.

In the next section we compare PLS and SVM regression based on the above

Table 1: Cross Validated Testing Set Results

Dataset	No of Obs.	Original No of Vars.	No of Vars.	q^2 PLS	q^2 SVMr
Aquasol	197	640	149	0.374	0.086
Blood/Brain Barrier	62	694	569	0.350	0.352
Cancer	46	769	362	0.438	0.623
Cholecystokinin	66	626	350	0.387	0.353
HIV	64	620	561	0.351	0.274
Malaria-1	76	1181	685	0.650	0.576
Malaria-2	76	1181	685	0.668	0.500

Table 2: Parameter Sets

Parameter	Values
ν	0.05, 0.1, 0.15, 0.2, 0.3, 0.35
C	100, 250, 500, 750, 1000, 1500, 2000, 3000, 4000, 5000
RBF σ^2	100, 200, 300, 400, 500, 650, 800, 1000, 3200, 5000, 6400

LP formulation (Eq. 2). Our SVM implementation reoptimizes the LP model for various parameter sets efficiently.

6 Experimental Results on QSAR Data

We conducted experiments based on the data provided by Electron Density-Derived Molecular Surface Area (EDDMSA) methodology [2] an improved version of Dr. Breneman's Transferable Atom Equivalent (TAE) methodology [3]. The datasets used in this section were created in an ongoing NSF funded Drug Design and Semi-supervised Learning (DDASSL) project (See `http://www.drugmining.com`). Basically, EDDMSA maps each compound to a larger set of spatially-resolved property variables of a type that has been shown to correlate with intermolecular activities. These variables are combined with the traditional topological variables to form the data for the QSAR analysis. Ten-fold cross validation was used in our experiments. We use exactly same training and test sets for both PLS and SVM regression. Table 1 summarizes the results. The malaria dataset has two possible response variables, each of them was modeled separately.

As we mentioned in the previous section, we screened our original data to remove the variables that have values out of the $\pm 4\sigma$ range. This is a common practice in commercial analytical tools used in chemometrics too. Thus we report two numbers of variables: The first one is before screening the data and the second one is after screening the data. This is a very primitive variable selection but our initial experiments showed that the screening never hurts and usually improves the results.

We used PLS procedure of SAS [13] with `RLGW` option for the PLS algorithm, since we have many factors. We picked the optimum number of factors based on the leave-one-out error for the each training set. We then predicted the test data based on the resulting PLS model. For the other options we used the SAS defaults.

In SVM regression experimental setup, one fold was reserved for the test data and another fold was reserved for the validation and eight folds are used for the training data. This is repeated for each possible validation fold. Thus for each test point, there are 9 predictions. As discussed above there is a high variance due to the use of the very small validation sets. Different sets results in greatly different models so we need to make the methods more robust. We use the average the outputs of 9 different model as our final prediction. This is a form of bagging [1]. This particular setup of the test, validation and training data also halves the total number of different models for a given parameter set from 90 to 45. We use same parameter sets (See Table 2) for all the datasets. Although the final number of different LP models is equal to 29700 ($45 \times 6 \times 10 \times 11 = 29700$), it takes a fraction of the time (less than five minutes for many of the datasets) to reoptimize them compared to the time to solve them from scratch.

We use the following statistics to compare two methods:

$$q^2 = \frac{\sum_{i=1}^{n}(\hat{y_i} - y_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

where $\hat{y_i}$ and $\bar{y}$ are the predicted and the mean value of the response variable y respectively. This is indeed equal to $1 - R^2$. But many chemometricians prefer the q^2 statistics. The lower the value of q^2, the better the model is. Experimental results are also given in Table 1. SVM regression performed as good or significantly better than PLS on all datasets except Cancer.

7 Conclusion

We have developed an effective methodology for application of Support vector regression to QSAR analysis. The first step is to scale the data and eliminate high variance features. To perform model selection, we utilize a linear programming SVM regression model that enables very efficient reoptimization after changes of model parameters. We perform an ordered grid search over a fixed set of parameters. The best model is picked based on the validation set error. Since there is high variance due to the small validation size, this process is repeated for different validation sets, and the resulting models are bagged or averaged. The final SVM regression model is very robust and outperforms PLS on most datasets. Currently we are working on enhancements to this SVM methodology including feature selection to further reduce the input dimensionality, more efficient model search methods, and applications to other drug design applications, e.g. high-throughput screening.

Acknowledgements

This work was partially supported by NSF IRI-9702306, NSF IIS-9979860 and NSF DMS-9872019. Many thanks to the Larry Lockwood and other members of the RPI DDASSL research group.

References

[1] L. Breiman. Bagging predictors. *Machine Learning*, 26(2):123–140, 1996.

[2] C. M. Breneman and M. O. Rhem. A QSPR analysis of HPLC column capacity factors for a set of high energy materials using electronic Van der Waals surface

property descriptors computed by the transferable atom equivalent method. *Journal of Computational Chemistry*, 18:182–197, 1997.

[3] C. M. Breneman, T. R. Thompson, M. Rhem, and M. Dung. Electron density modeling of large system using the transferable atom equivalent method. *Computers & Chemistry*, 19(3):161, 1995.

[4] R. Burbidge, M. Trotter, S. Holden, and B. Buxton. Drug design by machine learning: Support vector machines for pharmaceutical data analysis. http://www.rdg.ac.uk/~sas99acm/aisb00/abstracts/burbidge.

[5] T. G. Dietterich, A. N. Jain, R. H. Lathrop, and T. Lozano-Perez. A comparison of dynamic reposing and tangent distance for drug activity prediction. In *Advances in Neural Information Processing Systems, 6*, pages 216–223. Morgan Kaufmann, 1994.

[6] I. E. Frank and J. H. Friedman. A statistical view of some chemometrics regression tools. *Technometrics*, 35(2):109–148, 1993.

[7] ILOG Inc., Mountain View, CA. *CPLEX 6.5 User's Guide*, 2000. http://www.ilog.com.

[8] R. Kewley, M. J. Embrechts, and C. M. Breneman. Data strip mining for the virtual design of pharmaceuticals with neural networks. *IEEE Transactions on Neural Networks*, 11(3):668–679, 2000.

[9] S. M. LaValle, P. W. Finn, L. E. Kavraki, and J-C. Latombe. Efficient database screening for rational drug design using pharmacophore-constrained conformational search. In *Proceedings of 3rd Int. Conf. on Computational Molecular Biology, ReCoMB'99*, pages 250–259. ACM, 1999.

[10] W. Prusoff. The scientist's story. NY Times (03.19.2001).

[11] A.J. Smola, B. Schölkopf, and G. Rätsch. Linear programs for automatic accuracy control in regression. In *Proceedings ICANN'99, Int. Conf. on Artificial Neural Networks*, Berlin, 1999. Springer.

[12] A. Srinivasan and R. D. King. Feature construction with inductive logic programming: A study of quantitative predictions of biological activity aided by structural attributes. *Data Mining and Knowledge Discovery*, 3(1):37–57, 1999.

[13] R. D. Tobias. An introduction to partial least squares regression. Technical report, SAS Institute Inc., Cary, NC.

[14] S. Vijayakumar and S. Schaal. Locally weighted projection regression: An *O(n)* algorithm for incremental real time learning in high dimensional space. In *Proceedings of 17th Int. Conf. on Machine Learning, ICML'2000*, pages 1079–1086, 2000.