Modeling the Relative Binding Affinity of Steroids to the Progesterone Receptor with Probabilistic Neural Networks

Stefan P. Niculescu[1]* and Klaus L.E. Kaiser[2]

[1] TerraBase Inc., 3350 Fairview St., Suite 160, Burlington, Ontario, L7N 3L5, Canada
[2] National Water Research Institute, 867 Lakeshore Road., Burlington, Ontario, L7N 4A6, Canada

Abstract

The application of Probabilistic Neural Networks (PNNs) as modeling tools for the relative binding affinities of steroids to the progesterone receptor is investigated using a benchmark steroids data set. The results point towards the basic PNN with Gaussian kernel as the best model reported until present. The use of nonparametric statistics to confirm a model's potential to be used as a design/screening tool is also discussed.

1 Introduction

Modeling the relative binding affinities of steroids to the progesterone receptor is a very important research subject for various reasons. First, the pharmaceutical industry is interested in using such models as tools to design new drugs, for example more reliable contraceptives. Second, some of these compounds are passing through effluent treatment systems into the environment. Although nature neutralizes them through various mechanisms, such as UV-mediated or bacterial degradation, residual materials and their metabolites may cause effects on biota. Due to the cumulative effects of synthetic and naturally occurring estrogenic compounds, even those with low to moderate binding affinity could lead to dramatic changes in the health of populations as a result of endocrine disruption. Models for the relative binding affinities of chemicals to the progesterone receptor may be useful as screening tools for the assessment of substances' hazard for the environment. See van Helden et al. [1] and So et al. [2] for overviews on existing models. This paper focuses on investigating the potential use of Probabilistic Neural Networks as a modeling tool for the relative binding affinities of steroids to the progesterone receptor and pro-

vides a quantitative comparison with various models involving data reduction techniques.

2 Data

We use the benchmark data set created by van Helden et al. [1] on 56 steroids and ACT, i.e. the logarithm of their relative binding affinities (RBA) to the progesterone receptor. The model variables are all properties listed in Table 3 in the above quoted paper, except VOL_3 (no variability). We refer to the same paper for a complete discussion of the variables' interdependencies, the reasons for their inclusion in the data set, as well as the criteria used to select the compounds in the training and test sets.

3 Modeling Methodology

The modeling methodology discussed here is in essence a combination of neural networks and Bayesian statistics and provides a practical solution for the following mathematical problem: how to approximate the unknown distribution of a given population based on a learning set consisting of multivariate sample data, and without making any assumptions on the nature of the distribution itself. The theoretical solution of this typical Bayesian problem has been provided by Cacoullos [3] who extended to the multivariate case the one obtained by Parzen [4] for the corresponding univariate case. The multivariate estimator to be used has the general form

---

$$g(\bar{x}) = \frac{1}{n\sigma_1 \ldots \sigma_k} \sum_{i=1}^{n} \prod_{j=1}^{k} W_j \left[\frac{x_j - x_{i,j}}{\sigma_j} \right],$$

$$\bar{x} = (x_1, \ldots, x_k) \in D^k \subseteq \Re^k = (-\infty, +\infty)^k, \qquad (1)$$

where k represents the dimension of the vector space, D^k is the domain of the distribution, the learning set consists of all data vectors $\bar{x}_j = (x_{j1}, \ldots, x_{jk})$, $1 \le j \le n$, σ_1 to σ_k are scale parameters that control the size of the sphere of influence and would be set to smaller values for larger n, and W_1 to W_k are Parzen type univariate kernels. Meisel [5] set up the principles of how to use this estimator to perform both classification and mapping. Once the estimator is built, the predictions are generated via the well-known Bayes' Theorem. The real progress in implementing this paradigm has been made by Specht [6] who noticed that the associated computer algorithm may be split into a large number of simple separate procedures which can be run in parallel, a typical characteristic of a neural network. Specht built it and named it *Probabilistic Neural Network* in recognition for its theoretical roots. The so-called *basic* PNN corresponds to the particular case where $\sigma_1 = \sigma_2 = \ldots = \sigma_k = \sigma$ and $W_1 = W_2 = \ldots = W_k = W$. In that case, Eq. 1 reduces to

$$g(\bar{x}) = \frac{1}{n\sigma^k} \sum_{i=1}^{n} \prod_{j=1}^{k} W \left[\frac{x_j - x_{i,j}}{\sigma} \right],$$

$$\bar{x} = (x_1, \ldots, x_k) \in D^k \subseteq \Re^k = (-\infty, +\infty)^k. \qquad (2)$$

Obviously, the value of the scaling parameters resulting from the training of the PNN based on Eq. 1 will be more efficient than the one generated by the PNN based on Eq. 2. From the computational point of view, Eq. 2 is by far simpler and more manageable. Suppose we are interested in building a basic PNN model to map x_k as function of $x_1, x_2, \ldots, x_{k-1}$. The value of the sphere of influence parameter σ identified during the learning phase will differ from the ideal one by a multiplication factor. One may compensate for this difference by including as part of the model a convenient linear correction based on the training errors, altogether obtaining a better model. Note that such an approach is valid only for one variable mapping purposes. From all available Parzen type kernels the most convenient choice is the Gaussian kernel $W_G(t) = (2\pi)^{-1/2} \exp(-t^2/2)$, $t \in \Re = (-\infty, +\infty)$. For this kernel the influence of a sample point is steadily carried over a modest range and then smoothly tapers off and practically becomes zero after some distance. The effect is preventing sample points from exerting distant influences. The PNN architecture follows very precise rules. For instance, all PNNs based on Eq. 2 built to map x_k as function of x_1, $x_2, \ldots, x_{k-1}$ will have $k - 1$ neurons in the input layer, n neurons in the first hidden layer, $k - 1$ neurons in the second hidden layer (summation layer), and 1 neuron in the output layer. The activation functions are *de facto* conditional statistical estimators associated with the training population, and the only parameter subject to training is the sphere of

influence parameter σ. This explains why the learning phase of the PNN involves *only one pass* through the training data. At the level of the summation layer two quantities may be computed: the Bayesian estimate of the variable subject to mapping and its corresponding probability using the well-known Bayes Theorem. For the mapping problem, the estimate is simply handled to the output neuron. The associated probability is the key of using the PNN for classification purposes. The principle is very simple. Let A and B be two disjoint classes whose members are characterized by two unknown and distinct relationships $x_k = f_A(x_1, \ldots, x_{k-1})$ for class A, and $x_k = f_B(x_1, \ldots, x_{k-1})$ for class B. The problem is to classify a new individual to one of the two classes knowing only the values corresponding to the input variables in the two relationships. The solution involves two steps. The first one is to build for each of the two classes independent PNNs mapping x_k as function of $x_1, x_2, \ldots, x_{k-1}$ and trained on pre-specified subsets of cases known as belonging to that particular class. The second step is to use the two trained networks to perform the classification. For this purpose the input values corresponding to $x_1, x_2, \ldots, x_{k-1}$ are fed into both of them. The new case is classified to the class corresponding to the model generating the highest probability for the Bayesian estimate of the output variable x_k. The particular value of that estimate does not play any role in the classification process. The same principle is valid for more than two classes. A detailed description of the PNN computer algorithms is given by Masters [7]. For each of the model variables, a combination consisting of Z-transform and hyperbolic tangent functions was used for pre-processing purposes. The Z-transform moves the system of coordinates into the centroid of the training population and conveniently groups the data around it. The hyperbolic tangent is then compressing everything into $(-1, 1)$. This allows the PNN to take advantage and properly learn from extreme values and take care of possible outliers. The simplest way of judging a PNN's generalization performance is handled through external validation. See [8] and [9] for practical examples of PNN-type QSAR models based exclusively on molecular structure.

4 Results and Discussion

All models reported on here are based on the same groupings of compounds in the training and the external test sets as used by the best models identified by van Helden *et al.* [1] and So *et al.* [2]. This allows proper comparison of the results. The discussion involves two external test sets. The first, *Test*-vH, consists of the compounds labeled 44, 45, 47 to 51 and 53 to 56 in the van Helden *et al.* [1] data set. Compounds 46 and 52 were discarded by van Helden *et al.* (and termed "*structural outliers*") for the reason that their structural peculiarities were not properly represented by the structures of the compounds in the training set and the combination of input variables resulting from the Genetic Function Approxima-

tion (GFA) analysis of the data. A similar reason was used by So et al. [2] to base the discussion of their best model's performance (8-3-1 back-propagation neural network (BNN) identified via a genetic neural network (GNN)) on a second external test set, Test-S, consisting of all compounds in Test-vH except the compound labeled 56.

The model considered here is the basic PNN with Gaussian kernel trained on the given 43 compounds specified as training set by van Helden et al. [1]. The appropriate training correction generated by the training errors has been included in the model. The ACT values used for the computation were derived directly from the RBA values of Table 1 in van Helden et al. [1] in order to eliminate the discrepancies between some of the ACT values quoted in Table 3 and Table 6 of the same paper.

Comparison of the Pearson's correlation coefficients will allow positioning the PNN-based model with respect to other reported models built on the same van Helden et al. [1] data set.

The basic PNN with Gaussian kernel is based on recognition of the distributional characteristics of the population, while all the other models under discussion are based on minimizing distances and maximizing correlation coefficients. Consequently, in the idea that both the best van Helden et al. [1] and So et al. [2] BNN models were properly identified and trained, it is natural to expect from them to perform at least a little better than the PNN when it comes to compare the values of various statistical estimators associated with the predictive errors. The values of the maximum and minimum predictive errors for the PNN and the 10-5-1 BNN model identified by van Helden et al. [1] completely agree with this picture. Surprisingly, this is not the case with the rest of the indicators. Clearly the basic PNN with Gaussian kernel outperforms the 10-5-1 BNN model. The values of the Pearson's correlation coefficients corresponding to all models reported by So et al. [2] suggest the same picture of the PNN's superiority.

The training of the PNN involves only one pass through the training data and the only unknown parameter of the model is the sphere of influence parameter σ. This entails the possibility that the PNN may be subject to overfitting. In the same vein, creating the right BNN-type model requires the identification of the appropriate neural network architecture and finding the right moment to stop the learning (which involves repeated passes through the training data) corresponding to the best predictive capabilities. For the BNNs, both architecture (number of layers and number of neurons in each layer) and number of training cycles may be sources of overfitting/overtraining. The relative values of the Pearson's correlation coefficients corresponding to the training and the two test sets points towards the PNN with Gaussian kernel as the best balanced model. This comparison also suggests that the best models reported by van Helden et al. [1] and So et al. [2] are practically at par, and there are no sufficient performance reasons to claim superiority of either of these models over the other. Evaluating the idea that the PNN is overfitted, the overall poorer predictive performance of the two above quoted models may be interpreted as an indication that these two models are in fact subject themselves to more severe overfitting/overtraining than the PNN. However, if it is accepted that none of the models is overfitted/overtrained, then the only justification of poorer predictive performance for both best models reported in [1] and [2] must reside in the selection of the model variables which lack significant information to fully explain the data. The lack of superposition of the selections generated by the GFA and GNN, as presented in Table 3 in [2], strongly suggests the latter scenario.

In the conclusions section of their paper, van Helden et al. [1] express caution concerning the possible use of their model as tool for designing more active steroids. In contrast, So et al. [2] strongly suggest their model as being the one to be used for such purposes. An objective answer to the problem of recognizing a model's suitability as a tool for designing more active steroids may be provided by nonparametric statistics. The appropriate model must be characterized by the absence

Table 1. Performance comparison of the basic PNN with Gausian kernel and the 10-5-1 BNN model reported by van Helden et al. [1]

Neural network Set	PNN with Gaussian Kernel			10-5-1 BNN (GFA selected variables)		
	Training	Test-vH	Test-S	Training	Test-vH	Test-S
Minimum error	−0.421	−0.585	−0.585	NA	−0.549	−0.549
Maximum error	1.041	0.457	0.457	NA	0.412	0.412
Average error	−2.8E-06	−0.030	−0.022	NA	−0.089	−0.071
Standard deviation errors	0.258	0.316	0.332	NA	0.346	0.359
Sum square errors	2.802	1.008	0.995	NA	1.284	1.211
Average square error	0.065	0.092	0.099	NA	0.117	0.121
Pearson's r^2	0.863	0.766	0.769	0.880	0.590	0.599

NA = value not available.

Stefan P. Niculescu and Klaus L.E. Kaiser

Table 2. Pearson's r^2 correlation coefficient values for various models involving neural networks

Model	Training	Test-vH	Test-S
PNN (Gaussian kernel)	0.863	0.766	0.769
BNN using GFA descriptors[a]	0.880	0.590	0.599
GFA[b]	0.722	NA	0.506
Generalized Simulated Annealing[b]	0.860	NA	0.488
GNN[b]	0.880	NA	0.610

NA = Value not available.
[a] van Helden *et al.* [1].
[b] So *et al.* [2].

of trends in the errors produced on the external test set. In other words, the model must be impartial in judging the activity of all new compounds submitted to screening. As we do not have access to the predictions generated by the models considered by So *et al.* [2], we restrict the discussion only to the best van Helden *et al.* [1] 10-5-1 BNN model, and to the PNN-model with Gaussian kernel, using the *Test*-vH set. Let us denote by $\{x_i, 1 \leq i \leq n = 11\}$ the set of measured values for the compounds in *Test*-vH, and $\{\varepsilon_i, 1 \leq i \leq n = 11\}$ the set of corresponding model errors. The absence of trends between the two sets is equivalent with their statistical independence. For this purpose we build the associated Kendall's K-score statistics:

$$K = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \xi(\varepsilon_i, \varepsilon_j, x_i, x_j), \tag{3}$$

where n is the sample size and

$$\xi(a, b, c, d) = \begin{cases} 1 & \text{if } (a-b)(c-d) > 0, \\ 0 & \text{if } (a-b)(c-d) = 0, \\ -1 & \text{if } (a-b)(c-d) < 0. \end{cases} \tag{4}$$

For a two-sided test of the hypothesis H_0: independence, versus H_1: dependence, at the $\alpha \in (0, 1)$ level of significance:

$$\text{reject } H_0 \text{ if } K \geq k(\alpha_2, n) \text{ or } K \leq -k(\alpha_1, n),$$
$$\text{accept } H_0 \text{ if } -k(\alpha_1, n) < K < k(\alpha_2, n), \tag{5}$$

where $\alpha = \alpha_1 + \alpha_2, \alpha_1 > 0, \alpha_2 > 0$, and the constants $k(\alpha_i, n)$ are the solutions of the equations:

$$P\{K \geq k(\alpha_i, n)\} = \alpha_i, \tag{6}$$

The interested reader may find details on the subject as well as detailed tables listing the values of $k(\alpha, n)$ in Hollander and Wolfe [10]. The value of the associated Kendall score is $K = -15$ for the van Helden *et al.* [1] best model, and $K = -5$ for the basic PNN with Gaussian kernel. For both models under discussion $n = 11$ and $|K| \leq 25 = k(0.03; n)$. Consequently, at the significance level $\alpha = 0.06$, the hypothesis H_0

of independence cannot be rejected for any of the two models under discussion. In other words, with probability 0.94, both models are impartial in treating the compounds in the external test set. Of course, due to the very limited pool of data on which these models are based, common sense caution is recommended in using any of them as design/screening tools.

5 Concluding Remarks

The results of the present analysis clearly position the basic PNN with Gaussian kernel as one of the most efficient tools available at this time for the purpose of modeling the relative binding affinities of steroids to the progesterone receptor. This can be ascribed to its Bayesian nature and its mathematical simplicity. The right choice of parameters will continue to be a challenge for every data set subject to investigation. New and reliable data reduction methodologies capable of handling the variable selection for highly nonlinear models are needed for the simple reason that none of the existing ones performs this task properly. Using any QSAR model for design and/or substance screening raises the question on how to recognize the model's domain. At the present time there is no satisfactory answer to this extremely important problem either.

References

[1] Van Helden, S.P.. Hamersma. H.. and van Geerestein, V.J., Prediction of the Progesterone Receptor Binding of Steroids Using a Combination of Genetic Algorithms and Neural Networks, in: Devillers. J. (Ed.). *Genetic Algorithms in Molecular Modeling*. Academic Press. London 1996, pp. 159–192.

[2] So, S.-S., van Helden. S.P.. van Geerestein, V.J., and Karplus, M., Quantitative Structure–Activity Relationship Studies of Progesterone Receptor Binding Steroids, *J. Chem. Inf. Comput. Sci. 40*. 762–772 (2000).

[3] Cacoullos, T.. Estimation of a Multivariate Density, *Ann Inst. Stat. Math.* (Tokyo) *18*. 179–189 (1966).

[4] Parzen, E., On Estimation of Probability Density Function and Mode, *Ann. Math. Stat. 33*, 1065–1076 (1962).

[5] Meisel, W.. *Computer-Oriented Approaches to Pattern Recognition*. Academic Press, New York 1972.

[6] Specht, D.. Probabilistic Neural Networks, *Neural Networks 3*, 109–118 (1990).

[7] Masters, T.. *Practical Neural Network Recipes in C++*. Academic Press. San Diego, CA 1993.

[8] Niculescu. S.P.. Kaiser, K.L.E., and Schultz, T.W., Modeling the Toxicity of Chemicals to *Tetrahymena Pyriformis* Using Molecular Fragment Descriptors and Probabilistic Neural Networks. *Arch. Environ. Contam. Toxicol. 39*, 289–298 (2000).

[9] Kaiser, K.L.E.. and Niculescu, S.P., Modeling Acute Toxicity of Chemicals to *Daphnia Magna*: A Probabilistic Neural Network Approach, *Environ. Toxicol. Chem. 20*, 420–431 (2001).

[10] Hollander. M., and Wolfe, D.A., *Nonparametric Statistical Methods*, John Wiley & Sons, New York 1973.