QSAR/QSPR Studies by Kernel Machines, Recursive Neural Networks and Their Integration

Alessio Micheli[1], Filippo Portera[2], and Alessandro Sperduti[2]

[1] University of Pisa, Department of Computer Science,
[2] University of Padova, Department of Pure and Applied Mathematics

Abstract. We present preliminary results on a comparison between Recurrent Neural Networks (RecNN) and an SVM using a string kernel on QSPR/QSAR problems. In addition to this comparison, we report on a first attempt to combine RecNN with SVM.

1 Introduction

Recursive Neural Networks (RecNN) (see, for example, [8]) have been proposed as an extension of standard neural networks to the adaptive processing of structured data, such as chemical compounds. Especially in Chemistry, RecNN have been successfully applied to QSPR (quantitative structure-property relationships) and QSAR (quantitative structure-activity relationships) studies (see, for example, [1]). Recently, kernels for structured data have been proposed and some interesting results on their application is emerging. The aim of this paper is to start a comparison of the above approaches on specific QSPR/QSAR applications. In particular, a Recursive Cascade Correlation network is considered as representative for the RecNN, while for structured kernels we considered a string kernel [9]. For both approaches we report preliminary results obtained for QSPR on Alkanes (predicting the boiling point) and QSAR of a class of Benzodiazepines (predicting the non-specific activity (affinity) towards the Benzodiazepine/$GABA_A$ receptor). In addition to this comparison, we report on a first trivial attempt to combine RecNN with SVM, where SVM is used to perform a post-processing of the internal representations of RecNN.

2 QSPR/QSAR Tasks

Here we describe the QSPR/QSAR regression tasks we have considered.

QSPR Task: Alkanes

The problem consists in the prediction of the boiling point for a group of acyclic hydrocarbons (alkanes). The prediction task is well characterized for this class of compounds, since the boiling points of hydrocarbons depend upon molecular size and molecular shape, and vary regularly within a series of compounds, which means that there is a clear correlation between molecular shape and boiling point. Moreover, the relatively simple structure of these compounds[3] is amenable to very compact representations such as topological indexes and/or vectorial codes, which are capable of retaining the relevant information for prediction. For these reasons, standard multilayer feed-forward networks using "ad hoc" representations yield very good performances.

In order to perform a comparison with the considered methods, we decided to use as reference point the work described in [2] which uses standard multilayer feed-forward networks. For this problem, RecCC has been proved to be competitive with respect to *ad-hoc* techniques (see [1]). In fact, the obtained results compares favorably versus the approach proposed by Cherqaoui et al. [2], which represented the *state-of-the-art* results. The data set used in [2] comprised all the 150 alkanes with 10 carbon atoms: 135 compounds were used for training and 15 compounds for test[4].

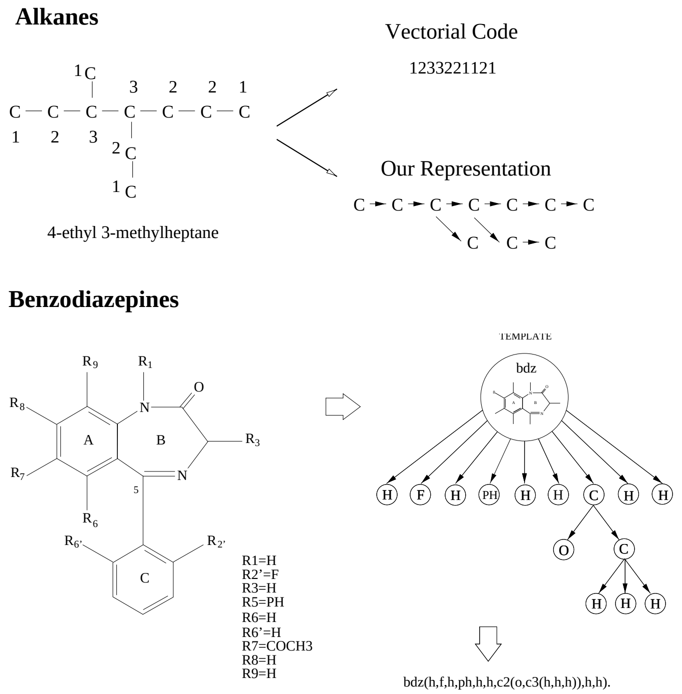

Fig. 1. Example of representation for an alkane and a benzodiazepine.

[3] No explicit representation of the atoms and bound type is required.

[4] It must be noted that Cherqaoui et al. use a vectorial code representation of alkanes based on the n-tuple code for the encoding of trees (see Fig. 1). So they represent each alkane as a 10 numerical components vector with the last components filled by zeros when the number of atoms of the compound was less than 10. The single component encodes the number of bounds of the corresponding carbon node.

QSAR Task: Benzodiazepines
The ability of predicting the biological activity of chemical compounds belonging to classes of therapeutical interest constitutes the major aspect of the drug design. Benzodiazepines, for example, has been extensively studied since the 70s, as this class of compounds plays the major role in the field of minor tranquilizer, and several QSAR studies have been carried out aiming at the prediction of the non-specific activity (affinity) towards the Benzodiazepine/$GABA_A$ receptor. The affinity can be expressed as the inverse of the logarithm of the drug concentration C (Mol./liter) able to give a fixed biological response[5].

A group of Benzodiazepines (Bz) (classical 1,4-benzodiazepin-2-ones) previously analyzed by Hansch et al. [3] through the traditional QSAR equations, was analyzed. The total number of molecules was 77, of which 5 are used as test set. The analyzed molecules present a common structural aspect given by the Benzodiazepine ring (see Fig. 1) and they differ each other because of a large variety of substituents at the positions showed in Fig. 1.

3 Neural Networks for Structures and the Tasks

Recursive neural networks (RecNN) [8] are neural networks able to perform mappings from a set of directed ordered acyclic graphs (DOAGs) (with labeled nodes) $\mathcal{I}^\#$ to the set of real vectors. Specifically, the class of functions which can be realized by RecNN can be characterized as the class of functional graph transductions $\mathcal{T} : \mathcal{I}^\# \to \mathbb{R}^k$, which can be represented in the following form $\mathcal{T} = g \circ \hat{\tau}$, where $\hat{\tau} : \mathcal{I}^\# \to \mathbb{R}^m$ is the *encoding* function and $g : \mathbb{R}^m \to \mathbb{R}^k$ is the *output* function. Specifically, given a DOAG $\boldsymbol{Y}$, $\hat{\tau}$ is defined recursively as

$$\hat{\tau}(\boldsymbol{Y}) = \begin{cases} \mathbf{0} \text{ (the null vector in } \mathbb{R}^m) & \text{if } \boldsymbol{Y} = \xi \\ \tau(s, \boldsymbol{Y}_s, \hat{\tau}(\boldsymbol{Y}^{(1)}), \ldots, \hat{\tau}(\boldsymbol{Y}^{(o)})) & \text{otherwise} \end{cases} \tag{1}$$

where a (*stationary*) τ can be defined as $\tau : \mathbb{R}^n \times \underbrace{\mathbb{R}^m \times \cdots \times \mathbb{R}^m}_{\text{o times}} \to \mathbb{R}^m$, $\mathbb{R}^n$ is the label space, the remaining domains represent the encoded subgraphs spaces up to the maximum out-degree of the input domain $\mathcal{I}^\#$, o is the maximum out-degree of DOAGs in $\mathcal{I}^\#$, $s = source(\boldsymbol{Y})$, $\boldsymbol{Y}_s$ is the label attached to the source of $\boldsymbol{Y}$, and $\boldsymbol{Y}^{(1)}, \ldots, \boldsymbol{Y}^{(o)}$ are the subgraphs pointed by s. A possible neural realization for τ is $\tau(\boldsymbol{l}, \boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(o)}) = \boldsymbol{F}(\boldsymbol{W}\boldsymbol{l} + \sum_{j=1}^{o} \widehat{\boldsymbol{W}}_j \boldsymbol{x}^{(j)} + \boldsymbol{\theta})$, where $\boldsymbol{F}_i(\boldsymbol{v}) = f(v_i)$ (sigmoidal function), $\boldsymbol{l} \in \mathbb{R}^n$ is a label, $\boldsymbol{\theta} \in \mathbb{R}^m$ is the bias vector, $\boldsymbol{W} \in \mathbb{R}^{m \times n}$ is the weight matrix associated with the label space, $\boldsymbol{x}^{(j)} \in \mathbb{R}^m$ are the vectorial codes obtained by the application of the encoding function $\hat{\tau}$ to the subgraphs $\boldsymbol{Y}^{(j)}$, and $\widehat{\boldsymbol{W}}_j \in \mathbb{R}^{m \times m}$ is the weight matrix associated with the jth subgraph space.

[5] In order to characterize the fixed response, the drug concentration able to give half of the maximum response (IC_{50}) is commonly used.

Concerning the output function g, it can be defined as a map $g : \mathbb{R}^m \rightarrow \mathbb{R}^k$. Specifically, in this paper, we use linear output neurons: $g(\boldsymbol{x}) = \boldsymbol{m}^T \boldsymbol{x} + \beta$, where $\boldsymbol{m} \in \mathbb{R}^m$ and β is the output threshold.

The architecture we have adopted for the experiments reported in this paper is Recursive Cascade Correlation (RecCC), which is described in detail in [8].

Molecular Structure Representation

The main requirement for the use of the RecNN consists in finding a representation of molecular structures in terms of DOAGs. The candidate representation should retain the detailed information about the structure of the compound, atom types, bond multiplicity, chemical functionalities, and finally a good similarity with the representations usually adopted in chemistry. The main representational problems are: to represent cycles, to give a direction to edges, to define a total order over the edges. An appropriate description of the molecular structures analyzed in this work is based on a labeled tree representation. In fact, concerning the first problem, since cycles are absent in the examined alkanes and they mainly constitute the common shared template of the benzodiazepines compounds, it is reasonable to represent a cycle (or a set of connected cycles, as in the benzodiazepines case) as a single node where the attached label carries information about its chemical nature. The second problem was solved by the definition of a set of rules based on the I.U.P.A.C. nomenclature system[6]. Finally, the total order over the edges follows a set of rules mainly based on the size of the sub-compounds. Examples of representations for alkanes and benzodiazepines are shown in Fig. 1.

4 A String kernel for trees

We start defining some notation to provide a kernel function for strings that will be applied to a string representation of chemical compounds (see Fig. 1). Let $\mathcal{A}$ a finite set of *characters* called *alphabet*. A *string* is a element $x \in \mathcal{A}^k$ for $k = 0, 1, 2, \ldots$. Let $|x|$ the length of x and let $v, x \in \mathcal{A}^k$, we say that $v \sqsubseteq x$ if v is a substring of x. Let $\text{num}_y(x)$ the number of occurrences of y in x. Let $\bar{\phi}(x)$ a function that maps $x \in \mathcal{A}^k$ in a feature space $\mathbb{N}^d$ with $d = \sum_{i=1}^{k} |\mathcal{A}|^i$, where each dimension represents a particular string of $\mathcal{A}^k$. Defining $\phi(x)_i = \text{num}_{s_i}(x)$, we can consider a dot product between vectors on this feature space [9]:

$$k(x, y) = \sum_{s_i \in \mathcal{A}^*} \phi(x)_i \phi(y)_i |s_i| \tag{2}$$

Note that, by definition, this is a kernel function since it represents a dot product in $\mathbb{N}^d \times \mathbb{N}^d$. Therefore, the kernel function $k(x, y)$ depends on the co-occurrences of substrings s_i both in x and in y. A match is then weighted with the length of

[6] The root of a tree representing a benzodiazepine is determined by the common template, while the root for alkanes is determined by the I.U.P.A.C. nomenclature system.

the common substring s_i. The function (2) can be computed in time $O(|x|+|y|)$ building the matching statistics with the Suffix Tree algorithm [9].

Having a kernel function that is able to compare variable length strings is helpful when input patterns are trees. In fact, given a tree T let $T' \sqsubseteq T$ if T' is a subtree of T. A possible kernel for trees is based on matching subtrees and has been proposed in [7]: $k(T_1,T_2) = \sum_{t_1 \sqsubseteq T_1, t_2 \sqsubseteq T_2} w_{t_1} \delta_{t_1,t_2}$, where $\delta_{t_1,t_2} = 1$ if $t_1 = t_2$ and 0 otherwise. A possible alternative is a string representation of trees and the use of the string kernel in eq. (2). Let T be a tree with l nodes and suppose that $\forall n \in T$, label(n) is a label associated to node n. Then for each node n in T assume that there exists a lexicographic order between the children nodes of n based on label(n). For each leaf node n, define tag(n) = label(n), for each internal node i let tag(n) = [label(n)tag($n1$) ... tag(n_c)] following the order defined on the children nodes of n. Thus, it can be shown [9] that the tag of the root node of T is a unique identifier for T and it can be computed in $O(l \log_2 l)$ time complexity.

5 Experiments

We performed experiments on 2 splits of the Alkanes dataset, called *a2* and *a9*, and the Benzodiazepines dataset called *Bz*. For these datasets, in Table (1) we have reported the results obtained by two instances of RecCC, denoted by the symbols a and b.

In the same table we have reported preliminary results of the string kernel as defined in eq. (2), calibrated on the test set to understand the potentiality of the approach that relies on a string representation of a molecule. In particular, the symbolic tree representation of the molecule used to generate the input for RecCC (by substituting symbols with binary vectors), is transformed into a string by the technique described in Section 4. These experiments were performed by integrating the string kernel in SVMLight 5.0 ([5]), that supports SVM regression.

We then explored the possibility to combine RecCC with a SVM. The first trivial attempt was to implement the output function $g()$ of RecCC by a SVM. Thus, we used the activity of the hidden units of an *already trained* RecCC (either a or b) as the input pattern for a SVM for regression with a RBF kernel. For each of the three problem instances we used both RecCC a and b. For this experiment, besides to SVMLight 5.0, we also used BSVM 2.04[7] [4]. We applied a 3-fold cross validation in the training set to fix the hyper-parameters C and γ and ϵ (the regression tube width) of the learning task. These results are denoted in Table 1 with *fair*. In addition, we report the results obtained by calibrating the learning algorithm on the test set.

As a final experiment of combination, we considered the same approach described above while using a dataset where each molecule is represented by the

[7] BSVM relies on a slight modification of the SVM primal quadratic model [6] that corresponds to a dual quadratic problem without general linear constraints. This explains the different results obtained with SVMLight and BSVM.

concatenation of the hidden activity of RecCCa and RecCCb. The results are shown in Table 2.

Table 3 reports a comparison of the results obtained with all the algorithms described in this paper. We did not report the results obtained by the concatenation of the hidden activity of RecCCa and RecCCb because it involves two indipendent RecCC models and so it would not be fair to compare these results with the ones obtained by a single model.

Preliminary results obtained by using a tree kernel based on the spectrum of a tree (i.e., number of shared subtrees), seems to give improved results with respect to the adoption of the string kernel. The tree kernel seems to be especially effective for dataset *a9*.

6 Conclusions

The aim of this paper was to start a comparison of Recursive Neural Networks (RecCC) versus kernel for structures (string kernel). The results for the string kernel seems to be worst with respect to the ones obtained by the Recursive Cascade Correlation network, however, these results, especially for the string kernel, are still very preliminary and need to be assessed. Moreover, a tree kernel will be considered in the future. More promising seems to be the preliminary results obtained by the simple attempts to combine RecNN with SVM, where SVM is used to perform a post-processing of the internal representations of the neural network for structure.

References

1. A.M. Bianucci, A. Micheli, A. Sperduti, and A. Starita. Application of cascade correlation networks for structures to chemistry. *Journal of Applied Intelligence (Kluwer Academic Publishers)*, 12:117–146, 2000.
2. D. Cherqaoui and D. Villemin. Use of neural network to determine the boiling point of alkanes. *J. Chem. Soc. Faraday Trans.*, 90(1):97–102, 1994.
3. Dimitra Hadjipavlou-Litina and Corwin Hansch. Quantitative structure-activity relationships of the benzodiazepines. a review and reevaluation. *Chemical Reviews*, 94(6):1483–1505, 1994.
4. C. Hsu and C. Lin. A comparison of methods for multi-class support vector machines, 2001.
5. T. Joachims. Text categorization with support vector machines: learning with many relevant features. In *Proceedings of ECML-98, 10th European Conference on Machine Learning*, pages 137–142, 1998.
6. O. L. Mangasarian and D. R. Musicant. Successive overrelaxation for support vector machines. *IEEE-NN*, 10(5):1032, September 1999.
7. M.Collins and N.Duffy. Convolution kernels for natural language, 2001.
8. A. Sperduti and A. Starita. Supervised neural networks for the classification of structures. *IEEE Transactions on Neural Networks*, 8(3):714–735, 1997.
9. S.V.N. Vishwanathan and Alexander J. Smola. Fast kernels for string and tree matching. In *NIPS 2002 Proceeedings*, 2003.

Problem	*max abs test err*	*mean abs test err*	C	γ	*epsilon-tube* [max abs train err]
a2(a) RecCC	0.0774	0.02714	max abs train err ≤ 0.080		
a2(a) SVMLight fair	0.0605	0.02426	1E4	0.001	0.015 [0.139]
a2(a) BSVM fair	0.0663	0.02532	1E4	0.001	0.015 [0.135]
a2(a) SVMLight	0.0587	0.01987	2E4	0.001	0.019 [0.143]
a2(a) BSVM	0.0590	0.02293	2E4	0.001	0.019 [0.110]
a2(b) RecCC	0.0647	0.02638	max abs train err ≤ 0.080		
a2(b) SVMLight fair	0.0583	0.02147	1E4	0.001	0.015 [0.076]
a2(b) BSVM fair	0.0615	0.02205	1E4	0.001	0.015 [0.074]
a2(b) SVMLight	0.0672	0.01919	2E4	0.001	0.0252 [0.074]
a2(b) BSVM	0.0689	0.01909	2E4	0.001	0.0252 [0.074]
a2 String Kernel	0.2879	0.06794	1	0.001	1.0E-8 [0.107]
a9(a) RecCC	0.3256	0.05112	max abs train err ≤ 0.080		
a9(a) SVMLight fair	0.3199	0.04847	1E4	0.001	0.01 [0.083]
a9(a) BSVM fair	0.3255	0.04822	1E4	0.001	0.01 [0.083]
a9(a) SVMLight	0.322	0.04809	1E4	0.001	0.009 [0.086]
a9(a) BSVM	0.3269	0.04844	1E4	0.001	0.009 [0.085]
a9(b) RecCC	0.2224	0.04819	max abs train err ≤ 0.080		
a9(b) SVMLight fair	0.2111	0.04744	1E5	0.001	0.048 [0.065]
a9(b) BSVM fair	0.2164	0.04804	1E5	0.001	0.048 [0.075]
a9(b) SVMLight	0.1846	0.04137	1E5	0.001	0.004 [0.099]
a9(b) BSVM	0.1766	0.04282	1E5	0.001	0.004 [0.123]
a9 String Kernel	0.1672	0.05070	1	0.001	1.0E-6 [0.476]
Bz(a) RecCC	0.06460	0.02542	max abs train err ≤ 0.040		
Bz(a) SVMLight fair	0.09019	0.03466	560	0.001	1E-9 [0.058]
Bz(a) BSVM fair	0.09048	0.03452	560	0.001	1E-9 [0.057]
Bz(a) SVMLight	0.05067	0.02320	9500	0.001	2E-3 [0.061]
Bz(a) BSVM	0.04612	0.02160	9500	0.001	2E-3 [0.062]
Bz(b) RecCC	0.08014	0.02307	max abs train err ≤ 0.040		
Bz(b) SVMLight fair	0.09075	0.03083	790	0.001	2E-5 [0.054]
Bz(b) BSVM fair	0.08956	0.03102	790	0.001	2E-5 [0.056]
Bz(b) SVMLight	0.07343	0.01986	5E4	0.001	2E-10 [0.028]
Bz(b) BSVM	0.07213	0.02220	5E4	0.001	2E-10 [0.033]
Bz String Kernel	0.08628	0.03770	0.1	0.001	1.5E-5 [0.014]

Table 1. Results on the *a* and *b* splits of datasets *a2*, *a9*, and *Bz*.

Problem	*max abs test err*	*mean abs test err*	*C*	*γ*	*epsilon-tube* [max abs train err]
a2 RecCC Best	0.0647	0.02638	max abs train err ≤ 0.080		
a2 SVMLight fair	0.0719	0.02748	18	0.001	0.015 [0.077]
a2 BSVM fair	0.0683	0.02844	18	0.001	0.015 [0.078]
a2 SVMLight	0.0324	0.01974	39	0.001	0.0031 [0.104]
a2 BSVM	0.0316	0.01842	39	0.001	0.0031 [0.104]
a9 RecCC Best	0.2224	0.04819	max abs train err ≤ 0.080		
a9 SVMLight fair	0.1633	0.05102	1E5	0.001	0.008 [0.009]
a9 BSVM fair	0.1593	0.05102	1E5	0.001	0.008 [0.009]
a9 SVMLight	0.1853	0.04618	96	0.001	0.0385 [0.045]
a9 BSVM	0.1848	0.04625	96	0.001	0.0385 [0.039]
Bz RecCC	0.08014	0.02307	max abs train err ≤ 0.040		
Bz SVMLight fair	0.07335	0.03814	140	0.001	0.008 [0.070]
Bz BSVM fair	0.07178	0.03802	140	0.001	0.008 [0.071]
Bz SVMLight	0.02386	0.01228	4.7E4	0.001	1E-6 [0.014]
Bz BSVM	0.01851	0.00835	4.7E4	0.001	1E-6 [0.020]

Table 2. Results obtained with feature concatenation of splits *a* and *b* on datasets *a2*, *a9*, and *Bz*.

Problem	*max abs test err*	*mean abs test err*	*C*	*γ*	*epsilon-tube* [max abs train err]
a2 RecCC Best	0.0647	0.02638	max abs train err ≤ 0.080		
a2 RCC+SVM Best	0.0689	0.01909	2E4	0.001	0.0252 [0.074]
a2 String Kernel	0.2879	0.06794	1	0.001	1.0E-8 [0.107]
a9 RecCC Best	0.2224	0.04819	max abs train err ≤ 0.080		
a9 RCC+SVM Best	0.1846	0.04137	1E5	0.001	0.004 [0.099]
a9 String Kernel	0.1672	0.05070	1	0.001	1.0E-6 [0.476]
Bz RecCC	0.08014	0.02307	max abs train err ≤ 0.040		
Bz RCC+SVM Best	0.07343	0.01986	5E4	0.001	2E-10 [0.028]
Bz String Kernel	0.08628	0.03770	0.1	0.001	1.5E-5 [0.014]

Table 3. Summary of the results obtained with several algorithms on datasets *a2*, *a9*, and *Bz*.