A Preliminary Empirical Comparison of Recursive Neural Networks and Tree Kernel Methods on Regression Tasks for Tree Structured Domains

Alessio Micheli

Dipartimento di Informatica, Università di Pisa, Pisa, Italia.
E-mail: micheli@di.unipi.it

Filippo Portera and Alessandro Sperduti

Dipartimento di Matematica Pura ed Applicata,
Università degli Studi di Padova, Padova, Italia.
E-mail: {portera, sperduti}@math.unipd.it

Abstract

The aim of this paper is to start a comparison between Recursive Neural Networks (RecNN) and kernel methods for structured data, specifically Support Vector Regression (SVR) machine using a Tree Kernel, in the context of regression tasks for trees. Both the approaches can deal directly with a structured input representation and differ in the construction of the feature space from structured data. We present and discuss preliminary empirical results for specific regression tasks involving well-known Quantitative Structure-Activity and Quantitative Structure-Property Relationship (QSAR/QSPR) problems, where both the approaches are able to achieve state-of-the-art results.

Key words: Kernel Methods, Kernels for Structures, Recursive Neural Networks, Learning in Structured Domains

1 Introduction

In recent years several researchers have started to consider the adaptive processing of structured data. This interest is motivated by two main reasons: *i)* several very important computational problems in bioinformatics, chemistry, document classification and filtering (just to name a few), require the use of some machine learning procedure to be properly treated because their complexity does not allow a formal and precise definition of the problem and thus no algorithmic solution to the problem is known; *ii)* in many of the above problems, the objects of interest are

more naturally represented via structured representations of different sizes, such as sequences, strings, trees, directed or undirected graphs, which retain all the structural information relevant for solving the task. Within this area there are two main streams of research which are relevant for the neural network community (for an overview see [15]): *a)* Recurrent and Recursive Neural Networks (e.g., see [11] for the basic theory and [1,2] for istances of recent developments on specific structures and applications in Bioinformatics); *b)* Kernel Methods for Structured Data (e.g., see [12]).

Alternative approaches to structured domain learning have been proposed within the field of symbolic approaches to machine learning, such as ILP [18,25,10], and within the field of probabilistic approaches (e.g., [9]). Recent related work has also involved the combination of generative models and kernel methods for sequences or structures (see [17,26]).

The main aim of this work is to start a comparison among the two approches described above for learning in domains constituted by trees, using the same basic assumptions for data representation. Specifically, here we discuss the differences between the two approaches and we report the results obtained for a (preliminary) empirical comparison of them on two representative regression tasks in the field of computational chemistry, namely, a Quantitative Structure-Activity Relationship (QSAR) problem and a Quantitative Structure-Property Relationship (QSPR) problem. The neural networks used for the comparison are Recurrent Cascade-Correlation networks [28,29], while as kernel methods, we have used a Support Vector Regression (SVR), with two different kernels: *i)* a kernel based on string matching, where a string represents a tree; *ii)* the tree kernel proposed in [7].

It should be stressed that, in the considered regression problems, Recurrent Cascade-Correlation networks have already compared favorably with respect to state-of-the-art standard approaches used in the QSPR/QSAR field [23,4], and for this reason we do not repeat here the comparison with traditional approaches.

This paper is an expansion of the the work presented in [21].

2 Regression of k-ary trees by Recursive NN and SVR with Tree Kernels

In this paper we focus on k-ary trees (in the following referred to as trees), which are rooted positional trees with finite out-degree k. In addition, we require that each node of a tree is associated to an element of a set L representing numerical or categorical variables, denoted as the label set. Examples of label set are given by a set of symbols, e.g. the alphabet used to generate a set of strings, or a set of real valued vectors which may represent, for example, the results of some measurement relating to the objects represented by each node of the tree. Let vert($\boldsymbol{t}$) be the set of

vertexes of a tree $\boldsymbol{t}$. Given a tree $\boldsymbol{t}$ whose vertexes are associated to elements of L, we use subscript notation when referencing the labels attached to vertexes. Hence $\boldsymbol{t}_v$ denotes the vector of variables labeling vertex $v \in \text{vert}(\boldsymbol{t})$. The void tree will be denoted by the special symbol ξ. Notice that, this class of trees T includes the classes of rooted ordered trees, sequences and vectorial data.

We consider in our framework the class of functions that can be characterized as the class of functional tree transductions $\mathcal{T} : T \to \mathbb{R}$, which can be represented in the following form $\mathcal{T} = g \circ \tau_E$, where $\tau_E : T \to \mathbb{R}^m$ is the *encoding* function and $g : \mathbb{R}^m \to \mathbb{R}$ is the *output* function. Thus we consider functions that take a tree as input and that return a real-valued output. Extension to multivariate output is trivial.

In the following we will describe the RecNN and the kernel approach within the above framework.

RecNNs rely on a recursive and adaptive realization of the encoding function τ_E. Free parameters of the model equips the realization of τ_E, allowing the learning algorithm to adapt the encoding function to the given regression task.

The kernel approach relays on a (implicit) map ϕ that allows to represent the input structure $\boldsymbol{t} \in T$ in some dot product space (the *feature* space). A similarity measure can be defined on the feature space, i.e. $K(\boldsymbol{t}, \boldsymbol{t}') =< \phi(\boldsymbol{t}), \phi(\boldsymbol{t}') >$ and large margin methods (SVR) can be used to learn the regression function. The function ϕ plays the role of encoding an input structure in a real space, i.e. $\tau_E(\boldsymbol{t}) = \phi(\boldsymbol{t})$. As a result, in this approach the function τ_E is chosen a priori by the kernel designer and it is crucial, for success, that the choice properly reflects the characteristics of the problem at hand.

2.1 Recursive Neural Networks

The recursive approach underlying Recursive neural networks (RecNN) is a feasible and natural way to process recursive-structure domains, where structured data such as variable-length sequences, trees and more in general directed ordered acyclic graph can be represented. Specifically, RecNN [29] are neural network models able to realize mappings from a set of k-ary trees [1] (with labeled nodes) T to the real set. In a RecNN, to encode a given tree $\boldsymbol{t}$, the following recursive definition of τ_E is used:

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

[1] More in general, RecNN can be applied to directed positional acyclic graphs (DPAGs).

where a (*stationary*) τ can be defined as $\tau : \mathbb{R}^n \times \underbrace{\mathbb{R}^m \times \cdots \times \mathbb{R}^m}_{\text{k times}} \rightarrow \mathbb{R}^m$, $\mathbb{R}^n$ is the label space, the remaining domains represent the encoded subtrees spaces up to the maximum out-degree k, $s = root(\boldsymbol{t})$, $\boldsymbol{t}_s$ is the label attached to the root of $\boldsymbol{t}$, and $\boldsymbol{t}^{(1)}, \ldots, \boldsymbol{t}^{(k)}$ are the subtrees pointed by s. A possible neural realization for τ is $\tau(\boldsymbol{l}, \boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(k)}) = \boldsymbol{F}(\boldsymbol{W}\boldsymbol{l} + \sum_{j=1}^{k} \widehat{\boldsymbol{W}}_j \boldsymbol{x}^{(j)} + \boldsymbol{\theta})$, 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 τ_E to the subtrees $\boldsymbol{t}^{(j)}$, and $\widehat{\boldsymbol{W}}_j \in \mathbb{R}^{m \times m}$ is the weight matrix associated with the jth subtree space.

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

The parameters of a RecNN are typically adapted by a gradient descent approach. The architecture we have adopted for the experiments reported in this paper is the Recursive Cascade Correlation (RecCC), a constructive implementation of a RecNN, which is described in detail in [4,29].

2.2 Support Vector Regression with kernel functions for trees

In this section we briefly describe the SVR method we employed for structures that have been applied to chemical compounds.

In [30] the Support Vector method for estimating real-valued function is described. It is based on the solution of a quadratic optimization problem that represents a tradeoff between the minimization of the empirical error and the maximization of the smoothness of the regression function. More formally, suppose that l inputs $(\boldsymbol{x_1}, \boldsymbol{y}_1), \ldots, (\boldsymbol{x}_l, \boldsymbol{y}_l)$ are given, where $\boldsymbol{x_i} \in \mathbb{R}^d$ are the input patterns, and $\boldsymbol{y}_i \in \mathbb{R}$ are the related target values of our supervised regression problem. The standard SVR model for a 1-norm ϵ-insensitive loss function is (see [8]):

$$\begin{aligned} &\min_{\boldsymbol{w},b,\boldsymbol{\xi}} \tfrac{1}{2}||\boldsymbol{w}||^2 + C(\boldsymbol{\xi}'\mathbf{1} + \boldsymbol{\xi}^{*\prime}\mathbf{1}) \\ \text{s.t. } \forall i \in [1 \ldots l] : & \\ &\boldsymbol{w} \cdot \boldsymbol{x_i} + b - \boldsymbol{y}_i \leq \epsilon + \boldsymbol{\xi}_i, \\ &\boldsymbol{y}_i - \boldsymbol{w} \cdot \boldsymbol{x_i} + b \leq \epsilon + \boldsymbol{\xi}^*_i, \\ &\boldsymbol{\xi}_i, \boldsymbol{\xi}*_i \geq 0 \end{aligned} \tag{2}$$

where the smoothness is controlled by $||\boldsymbol{w}||^2$, the norm of the regression function parameters, and the empirical error is linearly weighted. The solution of (2) can

be expressed just in terms of inner products of vectorial representations of input patterns. Considering our common transduction framework, we can write the SVR output function with $g(\boldsymbol{x}) = \sum_{i=1}^{l} \boldsymbol{m}_i K(\boldsymbol{x_i}, \boldsymbol{x}) + b$ where $\boldsymbol{m} \in \mathbb{R}^l$ is related to the dual optimal solution and a bias value b can be derived from the Karush-Kuhn-Tucker (KKT) conditions on optimality.

The chemical compounds we have studied are naturally representable as trees with discrete labels attached to nodes. In consideration of this, we have evaluated a string kernel, which has been applied to string representations of the trees, and a tree kernel able to directly deal with the tree representations. In the following we describe these two kernels.

2.2.1 A String Kernel

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. 2). Let $\mathcal{A}$ a finite set of *characters* called *alphabet*. A *string* is an 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 $y \in \mathcal{A}^k$, then we denote with $\text{num}_y(x)$ the number of occurrences of y in x. Let $\phi(x)$ a function that maps $x \in \mathcal{A}^k$ into a feature space $\mathbb{R}^d$ with $d = \sum_{i=1}^{k} |\mathcal{A}|^i$, where each dimension represents a particular string of $\mathcal{A}^k$. Defining $\phi(x)_i = \text{num}_{s_i}(x)\sqrt{|s_i|}$, we can consider a dot product between vectors in this feature space [31]:

$$K(x, y) = \sum_{s_i \in \mathcal{A}^*} \phi(x)_i \phi(y)_i \, . \tag{3}$$

Note that, by definition, this is a kernel function since it represents a dot product in $\mathbb{R}^d \times \mathbb{R}^d$. 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 the common substring s_i. The function (3) can be computed in time $O(|x| + |y|)$ building the matching statistics with the Suffix Tree algorithm [31].

This kernel can be applied to string representations of trees. In fact, a string representation of a tree can easily be obtained by introducing parentheses.

2.2.2 Kernel for Trees

Concerning the kernel operating directly on trees, we have chosen the most popular and used Tree Kernel proposed in [7]. It is based on counting matching subtrees between two input trees. Given an input tree $\boldsymbol{t}$, let $s^{(\boldsymbol{t})}$ be a subtree of $\boldsymbol{t}$ if $s^{(\boldsymbol{t})}$ is rooted in a node of $\boldsymbol{t}$ and the set of arcs of $s^{(\boldsymbol{t})}$ is a subset of connected arcs of $\boldsymbol{t}$ (note that, with this definition, leaves do not need to be necessarily included in the

subtree). We assume that each of the m subtrees in the whole training data set is indexed by an integer between 1 and m. Then $h_s(\boldsymbol{t})$ is the number of times the tree indexed with s occurs in $\boldsymbol{t}$ as a subtree. We represent each tree $\boldsymbol{t}$ as a feature vector $\phi(\boldsymbol{t}) = [h_1(\boldsymbol{t}), h_2(\boldsymbol{t}), \ldots]$ (see Fig. 1). The inner product between two trees under the representation $\phi(\boldsymbol{t}) = [h_1(\boldsymbol{t}), h_2(\boldsymbol{t}), \ldots\ h_m(\boldsymbol{t})]$ is: $K(\boldsymbol{t}, \boldsymbol{t}') = \phi(\boldsymbol{t}) \cdot \phi(\boldsymbol{t}') = \sum_{s=1}^{m} h_s(\boldsymbol{t}) h_s(\boldsymbol{t}')$. Thus this tree kernel defines a similarity measure between trees

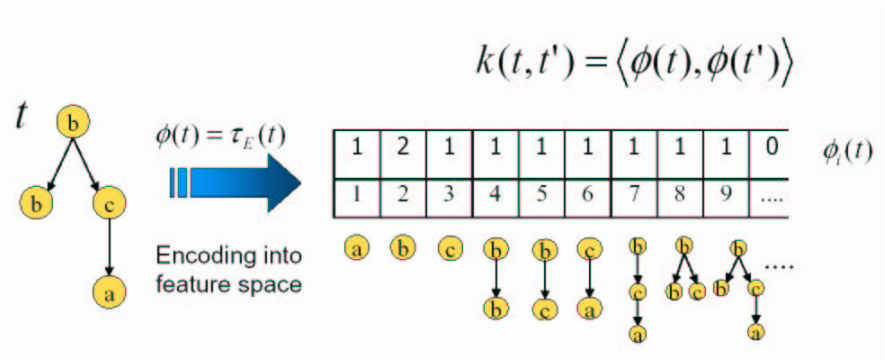

Fig. 1. Example of representation in feature space of a tree.

which is proportional to the number of shared sub-trees.

Experimental results showed that this kernel may weight larger substructures too highly, producing a Gram matrix with large diagonals. In [7], a method to dim the effect of the exponential blow-up in the number of subtrees with their depth is described. The proposal is to downweight larger subtrees modifying the kernel as follows: $K(\boldsymbol{t}, \boldsymbol{t}') = \sum_{s=1}^{m} \lambda^{\text{size}(s)} h_s(\boldsymbol{t}) h_s(\boldsymbol{t}')$ where $0 < \lambda \leq 1$ is a weighting parameter and size(s) is the number of nodes of the subtree s. The Tree Kernel can be calculated by a recursive procedure in $O(|NT| \cdot |NT'|)$ time where NT and NT' are the sets of nodes of trees $\boldsymbol{t}$ and $\boldsymbol{t}'$, respectively.

The kernel we use in this paper is slightly different from the one defined in [7], since we do not have constraints imposed by a grammar over the form of the subtrees to be considered when computing the kernel. Thus, we consider for computation all the possible subtrees.

3 QSPR/QSAR Tasks

Here we consider two paradigmatic instances of the regression problem defined on a structured domain, one for QSPR analysis, and one for QSAR analysis. Both problems have been previously faced by RecNN and favorably compared with respect to state-of-the-art standard approaches used in the QSPR/QSAR field [23,4].

The QSPR problem consists in the prediction of the boiling point for a group of acyclic hydrocarbons (alkanes). The data set used is described in [6,22] and com-

prised all the 150 alkanes with up to 10 carbon atoms, allowing to consider the problem of coping with structures of different sizes. The target values are in the range approximatively, in Celsius degrees, [-164 , 174].

The original aim of the application developed in [4] was the assessment of RecNN method by comparison with standard multilayer feed-forward networks using *ad hoc* vectorial representations of alkanes that yields *state-of-the-art* results [6]. The prediction task is well characterized for this class of compounds, since the boiling points of hydrocarbons depend upon molecular size and molecular shape (number and branching of carbon atoms in particular), 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 [2] (see Fig. 2) 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 [6], Cherqaoui et al. use a vectorial code representation of alkanes based on the *N-tuple* code for the encoding of trees (see Fig. 3). 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. In particular, the *N-tuple* code used in [6] is specific for the considered set of alkanes, as it provides a fixed dimensional vector for all the data, and it does not allow the representation of different atom symbols, for instance atoms or groups different from the carbon atom. Moreover, "ad-hoc" features, such are the number of carbon atoms and the branching of the molecular trees are explicitly reported in the representation. Hence, the information conveyed by the *N-tuple* code corresponds to the features we described as correlated to the boiling point property. As a result, the representation is efficient and the obtained predictions are very good for the prediction of the boiling points. However, when considering different classes of data, e.g. where various type of atoms occur, and different tasks, the representation assumptions could be useless. In particular, it is possibly required to design a different representation able to convey proper features related to the different predicted properties. Of course, a labeled tree representation can represent variable size-structures and can easily convey labeling information, which can be used to represent various type of atoms or chemical groups. Moreover, the topological aspect of the original tree is fully represented, therefore the topology information are not a priori selected on the basis of the specific target property. Hence, tree based representations result much more general for similar classes of acyclic compounds and they can be used for various tasks whithout the need to modify them according to the target property.

The QSAR problem considered here involves a class of chemical compounds be-

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

longing to a class of therapeutical interest: benzodiazepines. Several QSAR studies have been carried out aiming at the prediction of the non-specific activity (affinity) towards the Benzodiazepine/GABA$_A$ receptor. A group of benzodiazepines (Bz) (classical 1,4-benzodiazepin-2-ones) has been used for our experiments [23]. The total number of molecules is 72, of which 5 are used as test set. The target values range in [6, 9]. The analyzed molecules present a common structural aspect given by the benzodiazepine ring and they differ each other because of a large variety of substituents at the positions showed in Fig. 2. The original aim of the application developed in [4,23] was the assessment of RecNN by comparison with traditional Hansch QSAR approach, i.e. an equation-based approach using expert-selected physico-chemical properties as molecular descriptors [13].

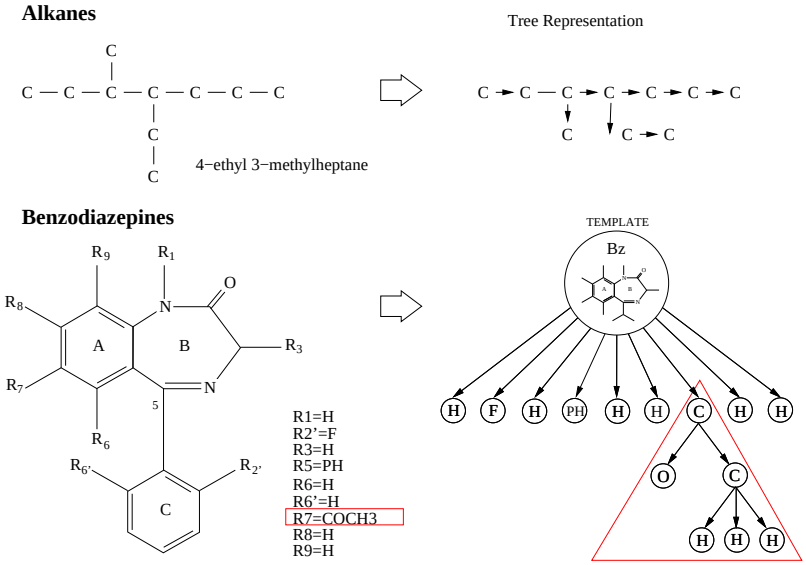

Fig. 2. Example of representation for an alkane and a benzodiazepine.

3.1 Molecular Structure Representation

An appropriate description of the molecular structures analyzed in this work is based on a labeled tree representation. Thus, both RecNN and Tree kernel can be applied, allowing us to preliminary compare them on a fair basis.

In order to obtain an unique structured representation of each compound, and their substituent fragments, as labeled positional trees (k-ary trees), we have defined a set of representation rules.

It is worth to note that alkanes (acyclic hydrocarbons molecules) are trees. In order to represent them as labeled k-ary trees, carbon-hydrogens groups are associated with vertexes, and bonds between carbon atoms are represented by edges; the root of the tree can be determined by the first carbon-hydrogens group according to the IUPAC nomenclature system [16] and the total order over the edges can be based

on the size of the sub-compounds.

In the case of benzodiazepines, the major atom group that occurs unchanged throughout the class of analyzed compounds (common template) constitutes the root of the tree. Note that, an alternative representation would have been to explicitly represent each atom in the major atom group (by a graph based representation). However, since this group occurs in all the compounds, no additional information is conveyed by adopting this representation. Finally, each substituent fragment is naturally represented as a tree once cycles are treated as replicated atom groups and described by the label information.

As a result the use of labeled trees (namely labeled k-ary trees) does not imply the loss of relevant information for these classes of compounds, which are representative of a large class of QSPR and QSAR problems. In particular, the representation of compounds is strictly related to the molecular topology and also conveys detailed information about the presence and types of the bonds, the atoms, and the chemical groups and chemical functionalities. Examples of representations for alkanes and benzodiazepines are shown in Fig. 2.

Summarizing, the representation rules (that are fully discussed in [23] and [4] for these sets of compounds) allows us to give an unique labeled k-ary tree representation of various sets of compounds through a conventional representation of cycles, by giving direction to edges, and by defining a total order over the edges. Since the rules are defined according to the IUPAC nomenclature, they retain the standard representational conventions used in Chemistry.

3.2 Discussion on the representation

The two tasks described here intentionally address two radically different problems in QSPR/QSAR with the aim of showing the flexibility of the proposed approaches to tackle different real-world problems defined on structured domain, while using the same computational approach. These two examples are meaningful and representative of a wider class of problems for the following reasons.

Simple structure often characterizes a wide set of problems where phisico-chemical properties would be predicted for organic compounds (QSPR tasks). In such studies, typically the problem is to find a compromise between the possibility to fully characterize the topology of the compound and the necessity to explicitly convey into the representation information concerning the occurrence of single atoms or groups. A tree representation with labeled nodes can naturally tackle both these problems, allowing to input much more information into the model than traditional approaches, e.g. topological indexes, group contribution methods, etc (see [4] for a short review). In particular, in the presented approaches the actual selection of the relevant information is left to the learning machinery.

For QSAR tasks it is very common to collect congeneric series of compounds which have the same mode of biological action, but with different quantitative levels, that medical-chemistry researchers would like to study. In these cases, it is typical to find a common template of the congeneric series and therefore to identify a nucleus-vertex where the structure can be rooted to.

Note that in both cases, the convention used in Chemistry, as for the standard IUPAC nomenclature, follows similar approaches to get unique representation of compounds.

However, it is worth to note that such examples are not intended to cover all the possible structures that can be found in the chemical domains, as the main aim of the paper is to computationally compare two different machine learning methods for structured domains on specific tree-structured domain tasks. A discussion on how to represent, in general, chemical structures deserves specific studies, as already done, although at an early stage of development, in [19,24,3]. In particular, in [19,24,3], it is shown that, with a proper treatment, complex chemical structures, including for instance steroisomerism (geometric isomerism or optical isomerism, i.e. Cis/trans and enantioners cases), cycles and position of cycles substituents, and even tautomeric configurations, can be represented in the proposed framework for the purpose of the QSPR/QSAR studies. Hence, for this specific studies, where the subject are the differences between two computational approaches, while the results should be interesting as an example of regression task on real-world structured data, the conclusion cannot necessarily be complete for the general problem of treatment of chemicals.

However, for the sake of comparison, we have used exactly the same representation of data for both the approches. Without regard to the specific assumption that can be used for the data representation, the two approches are applied under the same condition, thus the specificity of the representation with respect to the chemical domain does not undermine the comparison aim.

The main concept we would address in the current work is that the representation should not a priori exclude basic information such as the topological and the label content of the full structured representation of a chemical compound. In such way, the learning tool for structure domanin can exploit as much information as needed for the task at hand. The only goal of the representation rule introduced in [4] and exploited in the current applications is to find an unique representation of each molecule.

It is finally worth to note that for RecNN models, since the model is adaptive and it can modify the encoding process according to the training data (i.e. to the task), the arbitrariness that can result from the representation rules can be partially or totally compensated by the learning process. In particular for the RecNN approach, theoretical support to the generality of the encoding performed by the model, is given

by the approximation universal theorem showing that RecNN can approximate arbitrarily well any function from labeled trees to real values [14]. For the kernel approach the representation choice can be a stronger bias, as the similarity measures for data defined by the kernel should reflect the characteristics of the problem at hand.

4 General Comparison

In the following we outline the characteristics of the learning in structured domains approaches with respect to the standard vectorial approaches for QSAR/QSPR. Moreover, a general discussion on the characteristics of the RecNN and Kernel based methods for structures can be made prior to the experiments, on the basis of the unified presentation of the two approaches.

4.1 Standard Approach versus Structural Approach

As outlined in [23,5,19], the aim of QSAR/QSPR study is to find an appropriate function which, given a structured representation of a molecule, predicts for it a specific measurable property or biological activity. The function can be seen as a functional transduction from an input structured domain I, where molecules are represented, to an output domain O, such as the real number set, i.e. the property/activity values.

The QSAR/QSPR analysis can be decomposed in two sub-problems: *i)* the *representation problem*, i.e., how to encode molecules through the extraction and selection of structural features; *ii)* the *mapping problem*, i.e., the regression task usually performed by linear or non-linear regression tools (e.g., equational modeling, and feed-forward neural networks).

In traditional approaches the molecules are represented into a flat form, usually a fixed and finite dimensional vector, by an extraction of numerical features. For instance Hansch QSAR approaches for benzodiazepines ([13]) leads to the definition of molecular descriptors in the form of well-know measurable physico-chemical parameters, which are devised by the expert in the field. Various types of descriptors can be used, such as topological indexes, geometrical and electronic properties, or *ad hoc* vectorial code of the molecular connectivity. For instance, in the case study of alkanes taken from [6] a *N-tuple* code is used. So, even if the chemical graph is clearly recognized as a flexible vehicle for the rich expression of chemical structural information, the problem of using it in a form amenable directly to QSAR/QSPR analysis is still open. See [4,5] for reviews and details of traditional approaches in the view of transduction from structured domains.

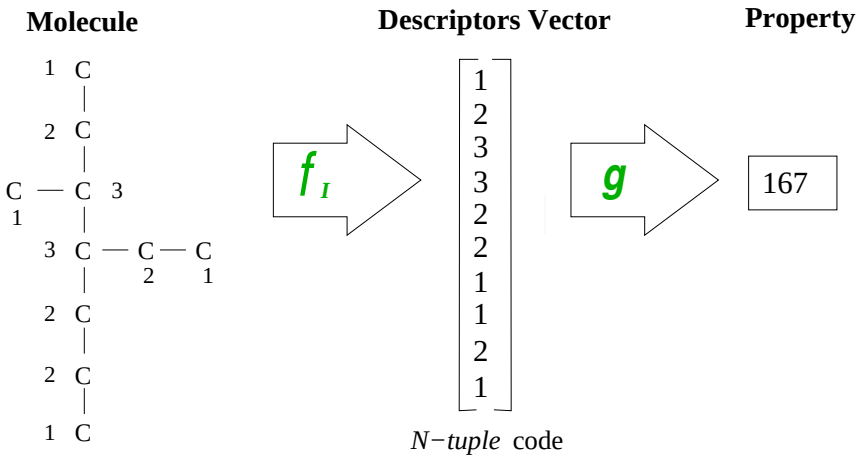

Fig. 3. Outline of the traditional approach to QSPR/QSAR (example on the QSPR of alkanes by *N-tuple* encoding).

As a results, in these approaches, machine learning machinery involves only the mapping function g, which can be realized by any known vectorial input model, e.g. multi-linear regression models, neural networks, SVM, etc. In Fig. 3 we report an outline of the traditional approach based on the instance of alkanes QSPR analysis, where f_I is a feature representation function solving the representation task, and g is the mapping function, a feed-forward neural networks in [6].

In the approaches proposed in this work, the model can take directly as input a structured representation of the molecules. As explained in Section 2 these structures take here the form of labeled trees. Thus, much more information can be conveyed into the model: the process can consider both the 2D structure topology (connectivity), the atom types, the chemical groups and functionalities occurring in each molecule, and deal with variable-size structures. The machine learning machinery realizing the transduction $\tau : T \rightarrow \mathbb{R}$ involves both the encoding function τ_E and the mapping function g (see Fig. 4). The construction of the features space is driven by an algorithmic technique, thus avoiding the use of hand-selected features. In particular, through different QSAR/QSPR tasks, we show how the generality and flexibility of a structured representation, allow us to deal with heterogeneous compounds and heterogeneous problems using the same approaches.

In the case of the kernel-based method, τ_E is realized by the kernel allowing implicit embedding of data into a high-dimensional features space. Since the space exploited by the kernel methods may have very high dimensionality (even infinite), the expressivity of such representation can be very high. However, the mapping performed by the kernel corresponds to the *a priori* definition of an encoding function. Since the kernel defines a similarity measures among data, it is crucial, to asses whether that similarity reflects the characteristics of the problem at hand. The function g is realized by a SVM.

In the case of RecNN, the encoding to numerical representation of chemical struc-

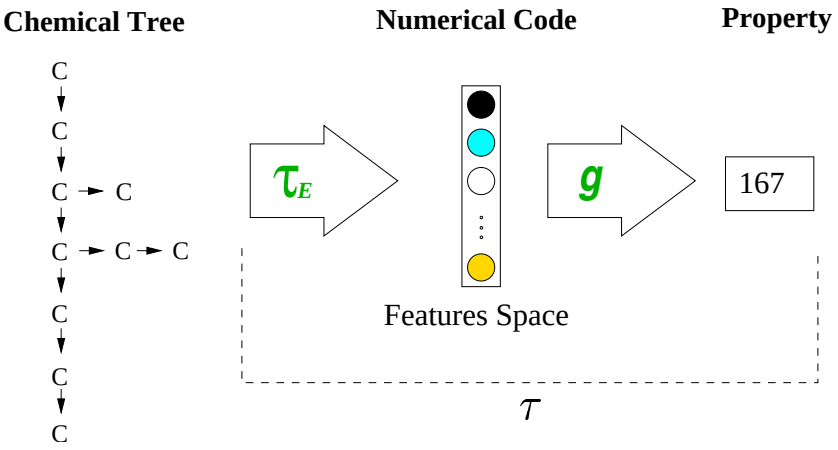

Fig. 4. Outline of the learning in structured domains approach to QSPR/QSAR (example on the QSPR of alkanes).

tures (τ_E) and the regression functions g are both realized by neural networks and they are learned together by the model. Hence, RecNN is able to *learn* a direct map between the input structured domain and the activity/property output space discovering numerical codes for the chemical structures which are optimized with respect to the prediction task. In other words, the similarity measure on data is adaptive in this case.

The output of the recursive neural network constitutes the regression output, while the internal representations of the recursive neural network (i.e., the output of the hidden units) constitute the neural implementation of the numerical descriptors returned by τ_E, i.e. a "focused" low-dimensional features space. It must be stressed, at this point, that the recursive neural network does not need to take as input a fixed-size numerical vector for each input graph, as it happens with standard neural networks typically used in QSPR/QSAR studies, because it is able to treat variable-size representations of the input graph. We may observe that the main difference between the traditional QSPR/QSAR scheme shown in Fig. 3 and the proposed new scheme reported in Fig. 4 applied to RecNN is due to the automatic definition of the τ_E function obtained by training the recursive neural network over the regression task. This implies that no *a priori* selection and/or extraction of features or properties by an expert is needed in the RecNN realization of τ_E.

4.2 Differences between Recursive Neural Networks and Kernels for Structures

As already pointed out, recursive neural networks learn the encoding function during training, while a kernel method inplicitly defines the encoding function before training. For kernel methods, thus, there is the risk to be unable to perform the computational task in the case the adopted kernel is not complete.

Learning in a recursive neural networks is performed via a gradient descent on a non-convex loss function (usually the mean square error), while for the kernel methods a constrained quadratic problem with a convex objective function must be solved. Because of that, recursive neural networks suffer the problem of local minima and training can be long and difficult in some cases. On the other side, kernel methods are particularly sensitive to hyperparameters, since the values they take basically define the feature space, and thus the hardness of the learning problem. Calibration of hyperparameters on the training set is a typical procedure used to overcome this problem. Of course, this leads to an increase in the training time. In structured domains, also the computation of the kernel can be computationally very heavy, especially when considering general graphs.

One big advantage of kernel methods is the theoretical basis which guarantees bounds on the generalization performance. Unfortunately, structures were the labels attached to vertexes are real-valued vectors, cannot be efficently dealt with kernel methods, since a structural kernel which is enough general to be useful in many cases would probably be computationally inefficient. In fact, fully general graph kernels cannot be efficiently computed [27].

5 Experimental Results

A systematic comparison of Recursive Neural Networks versus the kernel approach to structured domain processing is needed. In particular, we consider regression tasks where Recursive Neural Networks have already shown to be superior with respect to traditional approaches (see [23,4]), so to gain a better understanding of the suitability of tree kernels on the specific application domain. More in general, the main aim is to begin an assesment on the ability of the two approaches to function as general tools to deal with specific QSAR/QSPR problems without the necessity to develope a new computational model for slightly different problems.

The target values of the datasets are obtained by experimental procedures, so it is useful to fit them according to a maximal tolerance (ϵ_t) on the error. The used tolerance values are compatible with the experimental error and other QSPR/QSAR studies, i.e. $\epsilon_t = 8$ for the alkanes dataset and $\epsilon_t = 0.4$ for the benzodiazepines dataset.

For Recursive Neural Networks we decided to use Recursive Cascade-Correlation [4,29] and to stop training whenever the maximum absolute training error was below ϵ_t. The software we used for the kernel method is SVMLight 5.0 which follows a stop criterion based on the violation of the Karush-Kuhn-Tucker conditions of the computed dual solution. In fact, the criterion used by the solver disregards patterns with large error and with a related dual variable equal to C. So, the solution given in output can exhibit a maximum absolute training error that is above the experimen-

tal error. For the sake of comparison, for the SVR algorithm we implemented also a stop criterion where training is stopped when every support vector has an absolute error below ϵ_t. We evaluated two kernels, a String Kernel (2.2.1) and a Tree Kernel (2.2.2). In addition we evaluated a composition of an RBF function with both of them, obtaining the kernel

$$K_{RBF}(x, y) = e^{-\gamma(K(x,x)-2K(x,y)+K(y,y))} .$$

5.1 Measures of Performance

To measure the performance of the regression methods we used the average absolute error:

$$AAE = \frac{1}{|Set|} \sum_{(\boldsymbol{t}, f(\boldsymbol{t}))\in Set} |g(\boldsymbol{t}) - target(\boldsymbol{t})|$$

and the average squared error:

$$ASE = \frac{1}{|Set|} \sum_{(\boldsymbol{t}, f(\boldsymbol{t}))\in Set} (g(\boldsymbol{t}) - target(\boldsymbol{t}))^2$$

where Set is either the training or the test set. For the alkanes dataset the reported performances are averaged across different splits. Then we also report the standard deviation computed as:

$$\sigma = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(E_i - \mu_E)^2}$$

where n is the number of data splits, E_i is the AAE (or ASE) on the i-th split and μ_E is the mean AAE (or ASE) on the set of splits.

5.2 Settings for the Recursive Cascade-Correlation

Due to the large amount of parameters allowed by the Recursive Cascade-Correlation model, an initial set of preliminary trials were performed just to determine an admissible range for the learning parameters. However, no effort was done to optimize these parameters with respect to the two specific tasks: the main aim of the experiments was to show how Recursive Cascade-Correlation could deal with two completely different tasks using the same basic models. Due to the different result achieved by different random initialization for the connection weights, various trials were carried out for the Recursive Cascade-Correlation simulations and the mean values have been reported over five trials (alkanes) and six trials (Bz), respectively.

5.3 Settings for the SVR

As introduced in Section 4.2, for the calibration of SVR hyperparameters for alkanes, we shuffled the 150 patterns and we created 30 splits of 5 patterns each. The calibration involved a set of up to 4 parameters: the SVR constant C, the RBF kernel width γ, the Tree Kernel downweighting factor λ and the SVR regression tube width w. On the last 3 splits (each involving 145 training examples and 5 test examples) and for the case in which all the 4 parameters were involved, we applied a 3-fold cross validation based on a grid of $10 \times 5 \times 5 \times 9$ points generated by powers of 10 starting from 1×0.00001 for hyperparameters C and γ, steps of 0.01 starting from 0.01 for w, and steps of 0.1 starting from 0.1 for λ. For experiments involving less then 4 parameters, we just used the corresponding subset of the grid. We selected the parameter vector that gave the median of the best mean square validation error on the three splits and then we used these parameters for the final evaluation. For benzodiaziepines calibration we applied a 3-fold cross validation on the training set, based on the same parameters grid. We selected the parameter vector that gave the best mean square validation error.

5.4 Evaluation

The final evaluation for both Recursive Cascade-Correlation and SVR has been performed in the following way: a 10-fold cross validation has been performed for alkanes, while for benzodiazepines we evaluated the models on the original test set.

5.5 Results

The experimental results are reported in two tables (Table 1 and Table 2). In Table 1 we report the results obtained using the stop criterion which prescribes that the training process terminates when the maximum training absolute error is below a given tolerance ϵ_t. In this table, we report the results obtained for Recursive Cascade-Correlation ($RecCC$) and the four different SVRs, involving String and Tree kernels with (STK_{RBF} and TK_{RBF}) or without (STK and TK) subsequent application of an RBF kernel.

In Table 2 we report the results obtained by the four SVRs on the alkanes and benzodiazepines datasets using the SVMLight 5.0 termination criterion.

For all models we report the maximum absolute error on the training set (MAE tr, only for Table 2) and on the test set (MAE te), the mean absolute error on the training and test set (AAE tr, AAE te), the mean square error on the training and test set (ASE te, ASE te) and the hyperparameters obtained after the calibration

phase. Standard deviation is also reported for all results. Moreover, for SVR we have reported the average number of distinct support vectors (ANSV) for alkanes, and the number of distinct support vectors (NSV) for benzodiazepines. When considering neural networks, the corresponging entries report the average number of hidden units in both cases.

For the alkanes dataset $\epsilon_t = 8$ and for the benzodiazepines dataset $\epsilon_t = 0.4$.

¿From the experimental results it is possible to see that, when training is performed imposing a maximum error on the structures, Recursive Cascade-Correlation and SVR with Tree Kernel composed with an RBF kernel are almost equivalent, even if Recursive Neural Networks show a small advantage, especially for benzodiazepines. As expected, the use of a String Kernel does not reach the same performances which can be obtained by a Tree Kernel.

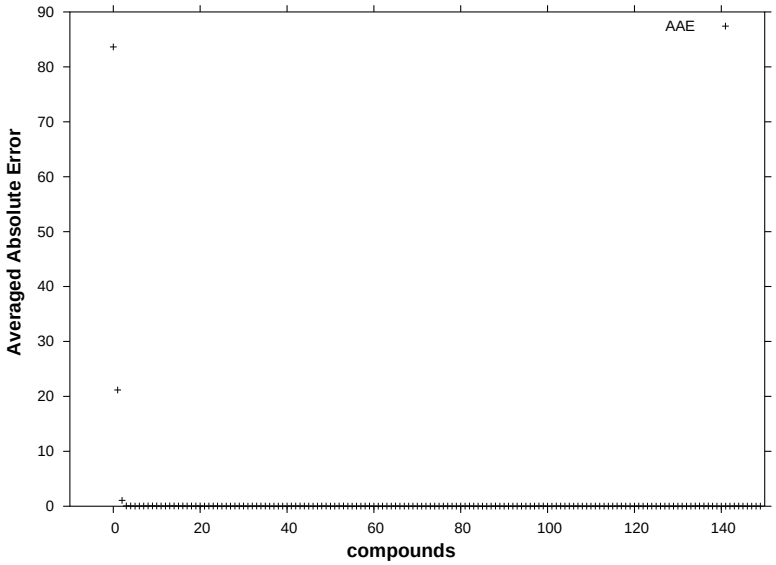

Fig. 5. Average error distribution for alkanes. Compounds are ordered on the x-axis in increasing size.

If the SVMLight stop criterion is used (Table 2), quite different results are obtained. First of all, it should be noted that for alkanes the maximum absolute error on the training set (*MAE tr*) reaches quite relevant values when using also the RBF Kernel, while the maximum absolute error on the test set (*MAE te*) decreases. Considering that both AAE and ASE are quite small, it is readly clear that just one input shows a quite relevant error. This is the case. In fact, looking at the distribution of the training errors averaged over the 10-fold cross validation splits [3] (see Fig. 5), it is observed that the error is concentrated on the smallest compounds (mainly on methane, which is represented by a single vertex and has a target value of -164, and ethane, which is represented by 2 vertexes and has a target value of -88.6).

[3] Actually, the average is computed over the 9 splits where the compound is present in the training set.

Moreover, the target values for the remaining 148 compounds are all values above -42.1 (which is the target value for propane). Thus it is clear that the calibration process has considered methane and ethane as outliers.

The dilemma here is whether this distribution of errors on the training set is acceptable or not. If the aim is just to produce a model for predicting the output values for the test set, then it is acceptable, however, if the aim is to model the whole set of trees, then this distribution of errors is not the most desiderable.

Another consideration for alkanes is that clearly the results for STK show a strong overfitting on the training set. This is due to the fact that in the calibration process C was equal or above 1. Since the output values for the String Kernel are quite large, smaller values for C should have been considered. In fact, the calibration returned the smallest value for C, i.e. 1. This problem can also be observed for STK_{RBF} on the benzodiazepines dataset, where the calibration returns $C = 1$.

When comparing the results versus the other stop criterion for SVR, a slight improvement on ASE is observed, especially for the Tree Kernel. In any case, the best results for benzodiazepines are obtained by the Recursive Cascade-Correlation.

6 Conclusion

We have shown, through the application of two different methods, that machine learning can properly deal with structured data, and that these approaches can be effective for real-world problems. Specifically, we have proposed for the first time the use of kernels for trees for QSPR/QSAR studies, and also a first comparison of the obtained results versus the ones obtained by neural networks for structures, i.e., Recursive Cascade-Correlation, which have already been proved to outperform traditional approaches on the considered regression problems.

Among the used kernels, i.e. a string kernel and a tree kernel, as expected, the best results are obtained by the tree kernel. The results for the string kernel are also worst with respect to the ones obtained by the Recursive Cascade-Correlation network. Recursive Cascade-Correlation networks seem also to perform slightly better than the tree kernel, even if it is difficult to compare the two approaches on a fair ground. Anyway, the experimental results clearly provide a further support to the hypothesis that for structured domains it is better to use methods able to deal directly with the structured nature of the domain.

The difference observed in the construction of the feature space by Recursive Cascade-Correlation and kernel based methods suggest that neural networks for structures can be considered a flexible tool to deal with unknown tasks because they are able to adaptively encode the structural information on the basis of the data and task at

alkanes					
	RecCC	STK	STK_{RBF}	TK	TK_{RBF}
AAE tr	2.15±0.12	2.52±0.41	2.16±0.32	3.70±0.21	2.43±0.24
ASE tr	7.85±0.88	9.44±2.68	7.79±1.83	19.32±2.35	8.99±1.81
MAE te	10.03±5.3	28.45±17.51	24.75±16.66	12.69±8.47	10.65±9.02
AAE te	2.86±0.74	7.03±1.84	5.49±1.64	4.70±1.06	2.93±0.92
ASE te	17.80±14.55	119.27±90.17	85.68±80.43	38.11±29.37	20.71±26.81
ANSV	140.1 (hidden units)	85.4	110.4	113.2	52.5
C	-	10	1E4	1E6	1E5
γ	-	-	1E-5	-	0.01
λ	-	-	-	0.16	0.16
w	-	0.02	0.00	0.00	0.02
benzodiazepines					
	RecCC	STK	STK_{RBF}	TK	TK_{RBF}
AAE tr	0.09	0.13	0.12	0.19	0.19
ASE tr	0.01	0.02	0.02	0.04	0.05
MAE te	0.61	0.78	0.76	0.91	0.88
AAE te	0.25	0.47	0.43	0.20	0.28
ASE te	0.11	0.29	0.25	0.17	0.17
NSV	19.7 (avg. hidden units)	44	44	46	47
C	-	1	10	100	1E6
γ	-	-	1E-5	-	1E-4
λ	-	-	-	0.04	0.04
w	-	0.01	0.01	0.02	0.02

Table 1
Results for the alkanes and benzodiazepines datasets obtained by SVR_c

hand. This deserves further research aimed at performing a deeper analysis on the comparison between the two approaches, focusing both on the different classes of functions that can be defined on structured domains and on different sets of experiments on real-world or properly designed artificial data.

Moreover, on the basis of the current discussion and results, a promising direction of research arises from the combination of the two approaches. Ensemble techniques can be used to this aim. Alternatively, a first attempt to combine Recursive Cascade-Correlation with SVM was preliminary introduced in [20], where a trained Recursive Cascade-Correlation model was used to compute the encoding function, and an SVM trained separately to implement the output function.

The ultimate aim should be the design of powerful models where SVM/SVR training is combined with the construction of an adaptive feature space focused on the problem at hand.

Acknowledgent

This work has been partially supported by MIUR grant n.2002093941_004

alkanes				
	STK	STK_{RBF}	TK	TK_{RBF}
MAE tr	0.09±0.01	78.58±16.63	2.18±0.02	14.85±3.84
AAE tr	0.03±0.00	0.73±0.15	1.68±0.03	1.12±0.04
ASE tr	0.00±0.00	49.94±15.69	3.19±0.08	4.29±0.75
MAE te	26.86±18.09	23.56±27.84	12.69±9.35	8.32±5.84
AAE te	5.92±1.60	4.66±2.45	3.82±0.97	1.86±0.46
ASE te	101.88±87.48	101.76±212.1	30.27±32.08	9.80±10.86
ANSV	134.6	134.0	90.0	134.9
C	1	10	1E5	1E4
γ	-	1E-4	-	0.01
λ	-	-	0.25	0.09
w	0.00	0.00	0.02	0.00
benzodiazepines				
	STK	STK_{RBF}	TK	TK_{RBF}
MAE tr	0.41	0.07	0.76	0.67
AAE tr	0.30	0.00	0.20	0.18
ASE tr	0.11	0.00	0.05	0.04
MAE te	0.84	0.72	0.75	0.75
AAE te	0.52	0.48	0.28	0.28
ASE te	0.31	0.26	0.14	0.14
NSV	35	67	49	46
C	1	1	10	1E5
γ	-	1E-4	-	1E-4
λ	-	-	0.04	0.04
w	0.04	0.00	0.02	0.02

Table 2
Results for the alkanes and benzodiazepines obtained by SVR.

References

[1] P. Baldi, S. Brunak, P. Frasconi, G. Pollastri, and G. Soda. Exploiting the past and the future in protein secondary structure prediction. *Bioinformatics*, 15(11):937–946, 1999.

[2] P. Baldi and G. Pollastri. The principled design of large-scale recursive neural network architectures-DAG-RNNs and the protein structure prediction problem. *Journal of Machine Learning Research*, 4:575–602, 2003.

[3] L. Bernazzani, C. Duce, A. Micheli, V. Mollica, A. Sperduti, A. Starita, and M. R. Tiné. Predicting thermodynamic properties from molecular structures by recursive neural networks. Comparison with classical group contributions methods. Technical Report TR-04-16, Università di Pisa, Dipartimento di Informatica, Pisa, October 2004.

[4] 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.

[5] A. M. Bianucci, A. Micheli, A. Sperduti, and A. Starita. A novel approach to QSPR/QSAR based on neural networks for structures. In L. M. Sztandera and H. M. Cartwright, editors, *Soft Computing Approaches in Chemistry*. Springer-Verlag, Heidelberg, March 2003.

[6] 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.

[7] M. Collins and N. Duffy. Convolution kernels for natural language. In *NIPS 14*, Cambridge, MA, 2002. MIT Press.

[8] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines and other Kernel-based Learning Methods*. Cambridge University Press, Cambridge, 2000.

[9] M. Diligenti, P. Frasconi, and M. Gori. Hidden tree markov models for document image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(4):519–523, 2003.

[10] S. Džeroski and N. Lavrač. *Relational Data Mining*. Springer-Verlag, Berlin, September 2001.

[11] P. Frasconi, M. Gori, A. Kuechler, and A. Sperduti. From sequences to data structures: Theory and applications. In *A Field Guide to Dynamic Recurrent Networks*, pages 351–374. Wiley-IEEE Press, 2001.

[12] T. Gaertner. A survey of kernels for structured data. *Newsletter of the ACM Special Interest Group on Knowledge Discovery and Data Mining*, 5(1):49–58, July 2003.

[13] D. Hadjipavlou-Litina and C. Hansch. Quantitative structure-activity relationships of the benzodiazepines. a review and reevaluation. *Chemical Reviews*, 94(6):1483–1505, 1994.

[14] B. Hammer. *Learning with Recurrent Neural Networks*, volume 254 of *Springer Lecture Notes in Control and Information Sciences*. Springer-Verlag, 2000.

[15] B. Hammer and B. J. Jain. Neural methods for non-standard data. In *Proceedings of ESANN 2004*, pages 281–292. D-side, 2004.

[16] IUPAC. *Nomenclature of Organic Chemistry*. Pergamon Press, Oxford, 1979.

[17] T. Jaakkola, M. Diekhans, and D. Haussler. A discriminative framework for detecting remote protein homologies. *Journal of Computational Biology*, 7(1,2):95–114, 2000.

[18] N. Lavrač and S. Džeroski. *Inductive Logic Programming : Techniques and Applications*. Ellis Horwood, 1994.

[19] A. Micheli. *Recursive Processing of Structured Domains in Machine Learning*. PhD thesis, Department of Computer Science, University of Pisa, 2003. TD-13/03.

[20] A. Micheli, F. Portera, and A. Sperduti. QSAR/QSPR studies by kernel machines, recursive neural networks a nd their integration. In B. Apolloni, M. Marinaro, and R. Tagliaferri, editors, *14th Italian Workshop on Neural Nets, WIRN VIETRI 2003*, volume 2859 of *Lecture notes in Computer Science*. Springer-Verlag, 2003.

[21] A. Micheli, F. Portera, and A. Sperduti. A preliminary experimental comparison of recursive neural networks and a tree kernel method for qsar/qspr regression tasks. In *Proceedings of ESANN'2004*, pages 293–298. D-side, 2004.

[22] A. Micheli, D. Sona, and A. Sperduti. Contextual processing of structured data by recursive cascade correlation. IEEE Trans. on Neural Networks. To appear., 2003.

[23] A. Micheli, A. Sperduti, A. Starita, and A. M. Bianucci. Analysis of the internal representations developed by neural networks for structures applied to quantitative structure-activity relationship studies of benzodiazepines. *Journal of Chem. Inf. and Comp. Sci.*, 41(1):202–218, January 2001.

[24] A. Micheli, A. Sperduti, A. Starita, and A. M. Bianucci. Design of new biologically active molecules by recursive neural networks. In *IJCNN'2001 - Proceedings of the INNS-IEEE International Joint Conference on Neural Networks*, pages 2732–2737, Washington, DC, July 2001.

[25] S. Muggleton. *Inductive Logic Programming*, volume 38 of *A.P.I.C. series*. Academic Press Ltd., London, 1992.

[26] L. Nicotra, A. Micheli, and A. Starita. Fisher kernel for tree structured data. In *Proceedings of the IEEE International Joint Conference on Neural Networks*, pages 1917–1922, July 2004.

[27] J. Ramon and T. Gartner. Expressivity versus efficiency of graph kernels. In *First International Workshop on Mining Graphs, Trees and Sequences*, 2003. http://www.ar.sanken.osaka-u.ac.jp/ washio/list/7.pdf.

[28] A. Sperduti, D. Majidi, and A. Starita. Extended cascade-correlation for syntactic and structural pattern recognition. In P. Perner, P. Wang, and A. Rosenfeld, editors, *Advances in Structural and Syntactical Pattern Recognition*, volume 1121 of *Lecture notes in Computer Science*, pages 90–99. Springer-Verlag, Berlin, 1996.

[29] A. Sperduti and A. Starita. Supervised neural networks for the classification of structures. *IEEE Transactions on Neural Networks*, 8(3):714–735, 1997.

[30] V. N. Vapnik. *Statistical Learning Theory*. Wiley, New York, 1998.

[31] S. Vishwanathan and A. J. Smola. Fast kernels for string and tree matching. In *NIPS 2002 Proceeedings*, 2003.