Pattern Recognition 48 (2015) 3749–3756

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

# Bridging from syntactic to statistical methods: Classification with automatically segmented features from sequences

J. Sidorova [a,b,*], J. Garcia [a]

[a] Lab for Applied Artificial Intelligence, Faculty of Computer Science, Universidad Carlos III de Madrid, Spain
[b] Department of Computer Science and Engineering, Blekinge Institute of Technology, Sweden

ARTICLE INFO

*Article history:*
Received 29 January 2015
Received in revised form
1 May 2015
Accepted 3 May 2015
Available online 14 May 2015

*Keywords:*
Syntactic pattern recognition
Grammatical inference
Feature segmentation
SMILES parser
Feature extraction

ABSTRACT

To integrate the benefits of statistical methods into syntactic pattern recognition, a *Bridging Approach* is proposed: (i) acquisition of a grammar per recognition class; (ii) comparison of the obtained grammars in order to find substructures of interest represented as sequences of terminal and/or non-terminal symbols and filling the feature vector with their counts; (iii) hierarchical feature selection and hierarchical classification, deducing and accounting for the domain taxonomy. The bridging approach has the benefits of syntactic methods: preserves structural relations and gives insights into the problem. Yet, it does not imply distance calculations and, thus, saves a non-trivial task-dependent design step. Instead it relies on statistical classification from many features. Our experiments concern a difficult problem of chemical toxicity prediction. The code and the data set are open-source.



## 1. Introduction

Statistical pattern recognition has a simple representation in the form of vectors allowing efficient ways to manipulate them, while syntactic pattern recognition has expressive representations, – graphs, strings, and so on, – but lacks object manipulation tools. Until recently, the syntactic and structural communities coexisted without much interaction. Yet, with the ever increasing difficulty of tasks in pattern recognition, more and more often the questions are asked: – *Can we have advantages of both paradigms?* – *Which are the trade-offs in such combinations?*

*Syntactic pattern recognition* can be used if there is a clear structure in the patterns and a *grammar* can be observed in a natural way. Forcing modeling on data, e.g. imposing linear ordering, hampers the performance [1]. Objects are represented by a variable-cardinality set of symbolic features.

Let there be $n$ different grammars $G_1, \ldots, G_n$, one for each recognition class $C_k$ $k = 1, .., n$. A *pattern* $p_x$ of an object $x$, – where $x$ can be a written digit, speech sample, protein sequence, etc. – must first be transformed to a sequence of terminal symbols, that is, smallest units. For example, a protein sequence as a string

$$p_x = ATTTGGGGCTTATATAT, \tag{1}$$

where $A$, $T$, $C$, $G$ are terminal symbols corresponding to the four nucleotides in the DNA. Examples of a recognition class $C_k$ form a training set $S(C_k)$

$$S(C_k) = \{p_{k_1}, p_{k_2}, p_{k_3}, \ldots\}, \tag{2}$$

and a grammar $G_k$ is sought, such that $L(G_k) \supseteq S(C_k)$. For a review of grammatical inference issues the reader is referred to [2,3].

There exist various *distance metrics* to measure similarity between patterns. Let $D(p_x, C_k)$ be some distance from a pattern $p_x$ to a class $C_k$. The (smallest) distance between an input pattern $p_x$ and a recognition class $C_k$[1] is

$$D(p_x, C_k) = \min\{D(p_x, p_k) | p_k \in L(G_k)\}. \tag{3}$$

In the literature, *three main approaches* to syntactic pattern recognition are typically singled out [4]:

- with an error-correcting parser,
- distance-based, and
- stochastic.

*An error-correcting parser* decides whether $p_x$ belongs to $L(G_i)$ or not. If $p_x$ belongs to $L(G_i)$, $x$ is assigned to category $C_i$, and it is rejected otherwise. *The distance-based scheme* computes a distance from $p_x$ to $L(G_k)$. If $D(p_x, L(G_i))$ is smallest among all the

* Corresponding author.
*E-mail address:* julia.a.sidorova@gmail.com (J. Sidorova).

[1] Or equivalently, between $p_x$ and $L(G_k)$.

http://dx.doi.org/10.1016/j.patcog.2015.05.001
0031-3203/

classes $C_1 \ldots C_n$, $x$ is assigned to category $C_i$. Here, a statistical component is often added, and the distances to recognition classes are the input to a statistical classifier, where C4.5 or the kNN are known to perform well and keep the classification process human readable. *Stochastic schemes* consist in adding occurrence probabilities to productions in the schemes defined above.

Obviously, object representation is crucial, and *graphs* would be ideal in many applications, but learning graph grammars is largely infeasible due to complexity issues,[2] instead graph embedding, e.g. [6,7], and kernel methods, e.g. [8,9], are used. For the research trend on graphs in pattern recognition, the reader is referred to [10]. *Strings* are suitable, since a regular or context-free grammar can be efficiently learnt and similarity measures are calculated. If the target language is regular, hidden Markov models (HMMs) have been used in many applications [11]. For example, they are the main-stream tool to discover chromatin states [12], or protein regions [13] with distinct biological functions. The problem is that HMMs treat sequences as one-dimensional strings of independent, uncorrelated symbols. Although computationally convenient, this assumption is not structurally realistic [14], because many phenomena have more complex structure than regular: natural language, palindrome structures in biology, and so on. Furthermore, once the target structure rises in terms of structural complexity from regular to context-free, one must make quite a number of task-dependent modeling decisions, and as a result applications become harder to design and reuse. Still, such efforts exist in optical character recognition [15], analysis of coronary artery images [16], in chemical biodegradability prediction [17,18], and some other.

*Statistical pattern recognition* has a simple representation in the form of vectors and efficient ways to manipulate them [19]. It has gained a much greater popularity than the syntactic paradigm. Yet, faced with ever growing difficulty of tasks, a recent tendency is to adapt ideas from syntactic methods. For example, in image understanding, ontologies are used for the loss function design: it is less of an error to take a cat for a dog, since both are animals, than a cat for a truck. In image tagging, structurally related features were shown to improve performance: if a ship has been detected, the probability for the sea should be high. In graph matching, structural information allows for constraint formulation: if a face is adjacent to a neck in one graph, it should be so in the other one, too. For an overview the reader is referred to [20]. Another idea proposed is to gain interpretability of predictive models in some creative task-dependent way, which often comes with a cost in recognition accuracy compared to black-box solutions or may require that the underlying linear model works well on the data set: for example, adding a heat map coloring technique to interpret linear support vector machine models [21].

This work, too, explores connections between the two paradigms, but our idea is different. In our previous work [18], we departed from the fact that there is a grammar for chemicals, very much like a natural grammar, and, we designed a syntactic pattern recognition scheme together with a procedure to search for important substructures in the grammars. In this submission, we propose to fill the feature vector with the counts of potentially important substructures. These substructures are automatically segmented, have an automatically chosen degree of structural abstraction and special statistical properties. The proposed *Bridging Approach* brings the following benefits:

1. The method's essential capacity is *to cope in the absence of expert knowledge*, that is, no indications with respect to which features to extract or where to look for them in the input sequence.
2. *It gives insights into the problem* in two respects. Firstly, the method works with a variable-length parsable input and finds the regions of interest in sequences with a suitable level of abstraction for their representation. Secondly, subsequent hierarchical vector-based feature selection and classification account for the domain's taxonomy.
3. *It is easier-to-implement* than a classical syntactic scheme, since it does not imply distance calculations. Therefore, it saves a non-trivial design step from the syntactic paradigm.

Our experiments concern a difficult problem of chemical toxicity prediction. Our parser processes molecules in the SMILES format, which is a string representation of a 2D molecular graph. From two sets of molecules with opposite properties $S(G_{\oplus})$ and $S(G_{\ominus})$, a predictive model is built with *the Bridging Approach*.

The rest of the paper is organized as follows. Section 2 explains how chemicals are represented as strings and how they are parsed. Section 3 explains the steps of *the Bridging Approach*. Section 4 covers the experiment. Finally, conclusions are drawn in Section 5. The SMILES parser and *the bridging approach* are available on request from the corresponding author. The database used for experiments is NCTRER DSSTOX at http://www.epa.gov/nheerl/dsstox/sdf_nctrer.html

## 2. Parsing chemicals

*The chemical language SMILES* was designed "*to represent molecular structure by a linear string of symbols, similar to a natural language*" [22]. A sequence in SMILES represents a molecular structure as a graph.

*Atoms*: Atoms are represented by their atomic symbols: C, Cl, N, O, etc. This is the only required use of letters in SMILES. Hydrogen atoms (H) are normally omitted, since valences make it clear where they are missing. For example, an atomic chain CCSCCCCC[3] is depicted in Fig. 1.

*Bonds*: Single bonds are usually omitted in SMILES. Double and triple bonds are represented by the symbols = and #, respectively, for example, in Fig. 2.

*Branches*: Branches are specified by enclosures in parentheses, as in Fig. 3.

*Cyclic structures*: Cyclic structures are represented by breaking one single (or aromatic) bond in each ring. The bonds are numbered in any order, designating ring opening (or ring-closure) bonds by a digit immediately following the atomic symbol at each ring closure. This leaves a connected noncyclic graph, which is written as a noncyclic structure, as in Fig. 4.

With the rules above almost all organic structures can be described as strings. For more details, the reader is referred to [22].

*A context-free parser based on the SMILES grammar* we developed creates a syntax tree from SMILES, see Appendix A for further details.

## 3. The bridging approach

Input is parsed structured data, the *Bridging Approach* will study it and build a predictive model based on its conclusions. Briefly, its steps are

[2] A problem of parsing non-trivial graph languages is PSPACE-complete or NP-complete. Defining graph-grammars generating languages with a polynomial membership problem is an open problem [5].

[3] Due to chemical convention in graphics, whenever a label on graph node is missing, it is C and a line segment represents a chemical bond.