ESANN'1999 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 21-23 April 1999, D-Facto public., ISBN 2-600049-9-X, pp. 63-68

A Benchmark for Testing Adaptive Systems on Structured Data

Markus Hagenbuchner
markus@uow.edu.au
University of Wollongong
2522 Wollongong, Australia

Ah Chung Tsoi
act@uow.edu.au
University of Wollongong
2522 Wollongong, Australia

Abstract

A number of adaptive methods capable of coping with structured data have emerged recently. Until recently, it was difficult to compare the performance of these methods as there are no universally accepted benchmark problems. As a result, we have developed a methodology to generate a benchmark problem sufficiently flexible to permit the simulation of a wide range of structured data learning problems, sufficiently fast to generate a set of patterns in a reasonable time, and sufficiently small to allow easy access to needed data. The benchmark described in this paper is an artificial learning task consisting of images that feature objects built through rules expressed by an attributed plex grammar. There are a number of advantages in utilizing this methodology. First, it can be well defined by using an attributed plex grammar. There is no need for the provision of a huge dataset of images as sets for training and testing can quickly be produced through a given grammar. But most importantly, this benchmark encapsulates some of the typical problems encountered in data processing of structured information. This paper illustrates this methodology by means of a traffic policeman problem. The patterns are used to generate data-trees as inputs for a typical adaptive learning algorithm. Preliminary tests show that some of these newly emerged adaptive learning algorithms perform very well compared to conventional methods.

1 Introduction

Currently, there is no universally accepted benchmark problems to validate and verify learning algorithms for data structures based on neural network concepts proposed recently by a number of research workers. This lack of a benchmark problem hampers the growing interest and further developments in this area. As one of the applications of structured data representation is image processing, there is considerable appeal in devising a benchmark problem based on image recognition. However, an inherent problem with image recognition benchmark is the difficulties involved in generating a set of images, as well as in the possible large amount of data storage required.

ESANN'1999 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 21-23 April 1999, D-Facto public., ISBN 2-600049-9-X, pp. 63-68

In this paper, we will describe a general methodology which can be utilized in the generation of image recognition benchmark problems, which can be used to validate and verify learning algorithms for data structures. This methodology is based on the concept of representing data structures using what is known as an attributed plex language, using an attributed plex grammar. The advantages of using such a methodology are that (1) it allows easy generation of a large number of image recognition problems, with very well defined parameters; (2) it requires relatively small memory storage; and (3) it captures some of the problems which face the tasks of adaptive processing of data structures. In order to illustrate this methodology, we will use an artificial learning task, viz., a traffic policeman, and to show how it can be applied readily.

The *traffic policeman* benchmark is an artificial learning task composed of images representing policemen while giving directions to the traffic. Policemen are composed of blocks of different shape, size, and color [1] which are properly combined by means of rules expressed using an attributed plex grammar. Some concepts can be defined in the world of the traffic policemen aimed at emphasizing either some features of the patterns or at recognizing an action.

Some learning tasks and experimental results are presented in section 4 and section 5. The software for generating images from an attributed plex grammar `polgen` is introduced in section 3.2 and the definition of the policeman benchmark is given in section 3.1. The following section gives a quick overview of the various grammars used. It is shown that attributed plex grammars are particularly useful for interconnecting and assigning properties to graphical objects.

2 Grammars

Common context free grammars [6] and popular notations are applied throughout this document and for the creation of the benchmark. These grammars are capable of modeling structural properties of patterns. However, they have difficulties in representing quantitative information such as the size of an object, textural parameters of regions, or orientation. Thus we decided to use attribute grammars [2] as a solution to this problem. The approach is to allow assignments of specific attributes to terminal symbols to describe and inherit size and color of objects. For the policemen benchmark we extend this grammar by the definition of global and local attributes. *Global attributes* are visible in all productions whereas *local attributes* are only visible within one production. Local attributes to an object A override its global attributes. Each object A has at least one global attribute namely its default value.

So far, the only possible relation between symbols in a string produced by a context free grammar is the concatenation relation. This schema is very restricted if higher dimensional patterns are to be described. A more effective approach is to incorporate relations into string representations which are more general than concatenation. The plex grammar, as introduced by Feder in 1971

[1] These are commonly known as primitives.

ESANN'1999 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 21-23 April 1999, D-Facto public., ISBN 2-600049-9-X, pp. 63-68

[3], introduces objects with an arbitrary number of attachment points for joining other symbols, thus allowing very general means of interconnection of primitive objects into complex objects. An object of this type is called an *n attaching-point entity* (NAPE) with NAPE $\in V$, where V is the vocabulary of the language, containing both the terminal set T, and non terminal set N. The ability to interconnect objects arbitrarily is particularly useful for this benchmark and is applied extensively.

Plex grammars are quite powerful and general in nature. They combine the power of string–, tree–, and web–grammars [4] as sub–cases. However, a drawback is that plex-productions could be quite cryptic. Furthermore, illegal productions are easily introduced but might be difficult to detect. For this reason methods for simplifying and normalizing plex-grammars have been suggested [5]

3 The policemen world

The world that we consider consists of a traffic policeman standing on a pedestal. The policeman is composed of blocks of different shape, color, and size. The shape of the blocks is predefined; color and size are variable. The policeman is in a certain state depending on how the blocks are assembled or modified. For example, if the policeman features a raised sign or two raised arms then his state is "*stop*", two lowered arms means "*go*", one raised left arm without a sign implies "*prepare for stop*". Other states can also be defined. States, for example, that rely on the size or color of body parts.

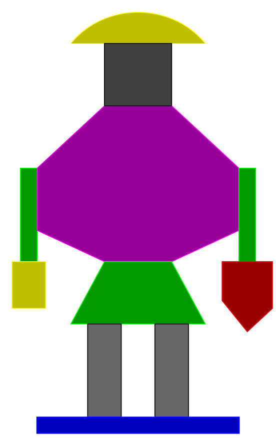


Image 1: A traffic policeman

3.1 Syntactic definition

The policeman world can nicely be described by means of an attributed plex grammar where a policeman can be composed of the following primitive objects:

$$N = \{\texttt{hat}, \texttt{face}, \texttt{mbody}, \texttt{arm}, \texttt{hand}, \texttt{sign}, \texttt{skirt}, \texttt{leg}, \texttt{pedestal}\}.$$

Interconnections between primitives are well defined through a plex grammar that also assigns modifying attributes to its primitive objects. The result are images similar to the one shown in image 1. A structural decomposition of these images is given in figure 1. It reflects the parsing tree as produced by the grammar.

ESANN'1999 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 21-23 April 1999, D-Facto public., ISBN 2-600049-9-X, pp. 63-68

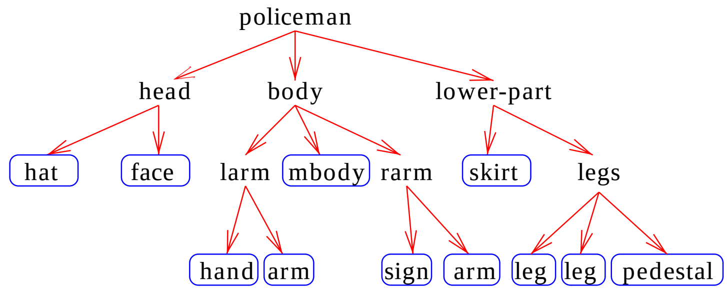


Figure 1: Structural decomposition of a traffic policeman

Every terminal symbol is represented by a graphical object in form of a pixmap file. It can be augmented with a set of *attributes*. Valid attributes assign a color or a scaling factor to each object. One of the main benefits of choosing pixmap data as terminal NAPEs is that there is no restriction to use just simple blocks as elements for the policeman grammar. They could also contain images of real world objects such as of a face of a person.

3.2 Creating Policemen

Policemen are created by utilizing a grammar as mentioned in previous chapters. The result is a collection of distinct images of equal resolution featuring policemen standing on a pedestal. An interpreter has been developed that accepts an attributed plex grammar as an input script file as well as certain command line options to control the number of policemen images produced or the desired resolution for each image. The interpreter *polgen* (short for **pol**icemen **gen**erator) creates an arbitrary number of policemen images from a given script file. It contains a large range of image manipulation functions as well as methods for syntax checking on the grammar, and an interpreter for producing images.

The number of different policemen[2] is controlled through the grammar. The number depends directly on the number of attributes stated as well as on the number of productions for each nonterminal.

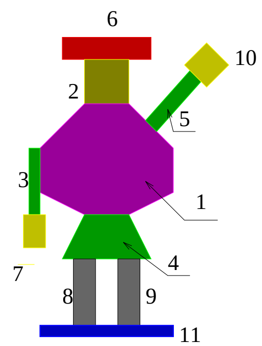
3.3 Graphical Representation

Numerous image preprocessing and feature extraction techniques have been developed to reduce the dimensionality for feature vectors. However, these methods mostly lack in an efficient representation of correlated data as the output is generally a feature vector of fixed or variable length. Graphical representations on the other hand are very informative as we can see in the example given in figure 2. It shows a tree representation of a policeman image. Nodes represent parts of the policeman. They are linked by arrows indicating the term

[2] The term 'different policemen' denotes to policemen produced by different parameters. There may be instances where identical policemen are produced from different parameters or productions.

ESANN'1999 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 21-23 April 1999, D-Facto public., ISBN 2-600049-9-X, pp. 63-68

"connected with". The root can be chosen arbitrarily. Such a representation not only utilizes methods for feature extraction as described before but also preserves correlation between extracted features.



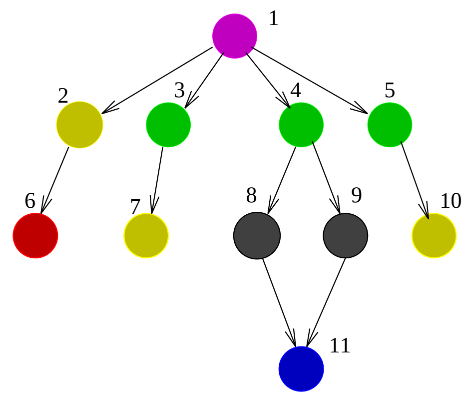


Figure 2: Graphical representation

A further advantage: Tree representations overcome problems with missing or wrongly extracted features more easily. For example, if an algorithm had failed to extract a hand of the policeman then the tree representation would still provide great similarities to the correct pattern. On the other hand, a tree representation can be less sensitive to scale and rotation. In the given example, the tree would be the same for policeman with a small body or for big policemen.

The traffic policeman benchmark is just an example of this general methodology of using attributed plex grammar to generate a whole class of images, suitable for validating learning algorithms for structured data. For example, one could imagine using the attributed plex grammar to describe the class of problems in Quantitative Structure Property Relationship (QSPR) and Quantitative Structure Activity Relationship (QSAR) in molecular chemistry [1].

4 Learning tasks

A large variety of learning tasks can be conceived in the policeman world. Tasks that can be designed arbitrarily difficult or easy. Here are some examples:

- Policemen with the hat of the same shape, color, size, ...
- Detect states such as the state "*stop*" for a raised sign or two raised arms.
- Types of policemen e.g. tall, medium, small, fat, slim.
- Policemen with missing elements e.g. no hat or with one leg only.
- Policemen generated by different grammars.
- Any combination of the former.

ESANN'1999 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 21-23 April 1999, D-Facto public., ISBN 2-600049-9-X, pp. 63-68

Tasks are considered to be *easy learning tasks* when structural information does not add any additional useful information to the given problem. Thus, finding all patterns from a given set of policemen that feature a raised left arm is an easy learning task. More interesting is to solve the disjunction-conjunction problem. E.g. find policemen wearing either a red or triangular Hat, have a missing left arm, and do not wear long pants.

5 Some Results

Experiments indicate that learning algorithms, such as the one described in [7], independent from architecture and updating method, are able to be trained successfully on this benchmark. On easy learning tasks (e.g. when classes are linearly separable) all tested systems are capable to perform at 100% recognition rate. A difference could be observed in the time needed for each individual network to converge. Common MLP networks required about half as many iterations as methods accepting structured inputs (such as BPTS [7]) to reach an error-level of less than 1%. Also, MLP networks utilized about 30 times less CPU-power. Least demanding in terms of CPU consumption was LVQ even so it commonly required about 10 times more iterations than MLP networks. These figures change dramatically when structured information is required to solve a given problem. Often, methods accepting structured information still produce reasonable results while other methods fail. However, this comes at the expense of more CPU-time per iteration.

References

[1] Micheli Alessio, Sperduti Alessandro, Starita Antonina, and Anna Maria Bianucci. Application of cascade correlation networks for structures to qspr and qsar. *submitted to Applied Intelligence*, 1998.

[2] Horst Bunke. Attributed graph grammars and their application to schematic diagram interpretation. *IEEE Trans. PAMI 4*, pages 574–582, 1982.

[3] T. Feder. Plex languages. *Info. Sciences 3*, pages 225–241, 1971.

[4] Rafael C. Gonzalez and Michael G. Thomason. *Syntactic Pattern Recognition : An Introduction.* Applied mathematics and computation ; no. 14. Addison-Wesley Pub. Co., Advanced Book Program, 1978. ISBN-0201029308.

[5] Marco Gori, Markus Hagenbuchner, and Ah Chung Tsoi. The traffic policeman benchmark. Technical report, Uni. of Wollongong, December 1998.

[6] John Edward Hopcroft. *Introduction to automata theory, languages, and computation.* Addison-Wesley series in computer science. Reading, Mass. ; London [etc.] : Addison-Wesley, 1979. ISBN-020102988X.

[7] A. Sperduti and A Starita. Supervised neural networks for the classification of structures. ftp://ftp.di.unipi.it/pub/Papers/perso/SPERDUTI/tr-16-95-revised.ps.gz, 1995.