An Efficiently Learnable Constructive Method for Graphs

Fabrizio Costa[1]

Introduction

Graph data structures allow us to model complex entities in a natural and expressive way. In the literature, several types of discriminative systems that can deal with graphs in input are known (e.g. recursive neural networks, graph kernels, etc), however, there are few generative or constructive approaches that can output graphs belonging with high probability to a desired distribution or class.

We argue that such systems are of great interest, and that novel and key problems in the field of Machine Learning (ML), and more generally in Artificial Intelligence (AI), can be addressed once such methods become viable. Let us digress a while and put things in perspective. Currently, the vast majority of models developed in the ML/AI research field are classifiers[2] (or regressors). These are used by practitioners of various disciplines (chemists, biologists, etc) mainly as proxies to replace expensive experimental measurements. In other words, ML tools are used to approximate hard to measure features (i.e. the biological activity of a molecule) on the basis of easy to measure features (i.e. the atomic composition and structure of a molecule). In the early days of AI and ML, the focus was on how to maximally exploit the information present in the handful of available data-points (mainly using domain knowledge to inject a strong bias both in the search strategy and in the hypothesis space). Nowadays however, the data bottleneck is disappearing in several domains (e.g. chemistry, biology, medicine). The dreaded consequence, as noted by [3], is that when sufficient data is available to cover the manifold, simple interpolation methods, such as the k-nearest neighbor technique, exhibit more than adequate predictive performance, rendering all other more sophisticated methods irrelevant. Further progress and increased ingenuity in experimental approaches are likely to exacerbate the issue. Nowadays, large scale and high-throughput experimental techniques can address directly many questions whose answer could previously only be approximated by computational methods (e.g. large screening to assess the biological activity of thousands of molecules, the assessment of large protein-protein interaction networks). In fact, it seems inevitable that in many fields the experimental approach will supersede the computational modeling strategy, and that it will do so at an accelerating rate[3].

[1] Bioinformatics Group, Department of Computer Science, University of Freiburg, Georges-Köhler-Allee 106, D-79110 Freiburg, Germany. Email: costa@informatik.uni-freiburg.de

[2] There are naturally many other types of contributions which do not fall under the classification type: feature selection, clustering and data mining, structured output prediction, to name a relevant few.

[3] To understand why, consider that the cost of several high-throughput techniques is decreasing at a faster rate than the cost for electronics predicted by Moore's laws.

The Challenge

One way for ML and AI to contribute to the advancement of other scientific disciplines, is to seek a type of task that cannot be easily solved by enhanced experimental techniques alone: one such task is the *learnable design problem* (LDP)[4]. A design (or generative) problem is formulated as the task to output instances belonging (with high probability) to a desired class. A learnable design or generative problem requires in addition that the class be (efficiently) modeled or learned starting from a finite sample of representatives instances.

Disregarding, initially, any limits derived from encoding and representational issues, we can list a range of interesting LDP application cases.

Given a set of game types ranked by a single player preference, one could automatically design a series of ever novel personalized games.

Given a set of molecules that have a desired biological activity one can design novel molecules that exhibit the same activity.

Given a set of bacteria whose metabolism byproduct is of utility (e.g. fermentation, carbon dioxide fixation, etc) one could design novel metabolic networks that have improved or multiple properties of interest.

In order to formulate these types of learnable design problems we need to address three main issues: 1) define a rich and expressive enough formalism to encode complex entities; 2) gain access to a sufficiently large representative set of design cases (and possibly also to a set of counter-examples); and finally 3) develop techniques that can efficiently learn the design principles and subsequently generate candidate design solutions.

We contend that we now have all the ingredients to start tackling the LDP. An adequate candidate for point 1) is the hyper-graph representation formalism, with which to encode arbitrary discrete entities and their relations. Point 2) is increasingly less critical given the current explosion of data (even of structured and relational type) available in machine readable formats. Point 3) is the subject of this work and is detailed in the following.

Related Work

The design problem is present, under different names and with important differences, in several research areas.

Operations research is often concerned with the identification of problem solutions (which can be at times regarded as design solu-

[4] We note that there exist however exceptions, i.e. a design problems that can be tackled in a pure experimental way; one such case is the design of *small molecules* when addressed by a *combinatorial chemistry* approach. In this case, the experimental approach materializes in hardware a typical software optimization algorithm.

tions) that satisfy some optimality condition (i.e. the cheapest circuit layout, the minimal length travel path). Traditionally however it assumes both the the objective function as well as the constraints to be given in an explicit form rather. (i.e. it does not include a learning stage)

Genetic algorithms and other derivative-free optimization techniques are also used to solve certain design problems (e.g. in mechanical and civil engineering fields), but they usually encode instances with fixed length vector representations and are thus not immediately suited for dealing with complex data.

The approaches that most explicitly address the LDP are grammar induction techniques. However, methods for learning expressive grammars are still not well developed, even in the case of simple string languages, let alone more complex domains like graphs or hyper-graphs. Graph grammars have been actively studied since the late 1960's, but few papers have dealt with their stochastic versions (needed to guarantee robustness in the presence of noise and outliers) and even fewer have dealt with the task of learning the grammar from a finite sample[5].

A few graph grammar induction methods, though, have been implemented. Usually they are not robust w.r.t. outliers or other form of structural noise [6], they tend to have high computational costs (i.e. they do not scale well to tens of thousands of graphs), they assume the structure of the grammar is given [9], or they are limited to context-free type of grammars, which possess nice properties of decidability, but suffer from limited expressive power [8].

The Quality Assessment Issue

A crucial issue, in the learnable design problem, is the model quality assessment. While in classification problems it is easy to devise metrics to compare the predicted class to the available *true* class, in the case of newly designed instances the issue becomes more complex. First of all, we have to deal with the lack of true class assignments. Resorting to an *oracle* can be difficult or at times impossible[6].

To tackle this issue we propose a simple yet effective strategy: given a set of examples and counter-examples, we induce the generative model, we train a binary classifier only on newly designed candidates, and we compute the predictive performance on a test set of known instances; the result is compared to the performance obtained by the same type of discriminative learner when trained on the original set. If the newly constructed instances belong to the same concept class, then both models should[7] perform equally well on the same test set. The size of the deviation can then be used as a measure of similarity between the true concept and the learned one.

Method

Here we propose a supervised constructive approach that, differently from current graph grammar induction techniques, is context-sensitive, is robust to outliers and is computationally efficient, with linear time complexity in both the model induction and the candidate generation phase.

The key notion is derived from that of *substitutability* [4]. Recently Clark and Eyraud [1] showed that certain types of grammars can be identified in the limit from positive data alone using the congruence classes of the language. The idea is to assume that all substrings that *always* occur in the same context (i.e. that are *congruent*) belong to the same implicit category and can therefore be substituted in the generative phase. We extend this notion in two ways: 1) we upgrade it from strings to graphs; and 2) we allow a local notion of context. More specifically we restrict the substitutable graphs to *neighborhood subgraphs* of given radius R, i.e. graphs induced by a root vertex v and all vertices that are within distance R from v. We then define the notion of *interface* as the difference between two co-rooted neighborhood subgraphs of different radii. The interface constitutes the local context for the inner co-rooted neighborhood graph that we call *core*. We call interface *thickness* T half of the difference of the two radii.

The model induction phase consists in the enumeration of all possible cores and their correspondent interfaces, rooted in all vertices of all positive instances. We then train a robust and fast graph kernel SVM discriminative model (introduced by Costa *et al.* in [2]) on a dataset containing both positive and negative examples. Finally, the construction of candidate graphs is achieved as an iterated substitution of congruent cores (i.e. cores with matching interface). The resulting graphs are evaluated in their entirety by the SVM; a simple beam search strategy is applied to control the size of the generated set. Initial findings support the idea that no sophisticated technique is required in order to escape local minima, as, given the size of the substituted subgraphs, a very large search space is in fact explored at each step. Finally, we show how the quantity $(R-T)/(R+T)$ can be regarded as the generative procedure's *creative tendency*, since a small radius and a large thickness result in very conservative substitutions supported by large contexts, while large radius and small thickness result in non-constrained substitutions.

We present encouraging experimental results in the chemoinformatics domain. Here the design task is the *de novo* construction of molecular graphs [5] that exhibit toxic properties [7]. The initial findings show a significant increase in predictive performance when we add 4K generated molecules to the initial dataset of comparable size, corresponding to a 14% relative error reduction for the area under the precision recall curve. Finally, we report how the generative procedure re-creates up to 5% of the original test set molecules.

[5] The main area of research has been rather the study of the generated language properties or the graph recognition problem given a specific type of grammar.

[6] E.g. assessing the activity of a newly proposed molecular graph requires the extremely complex and expensive step of molecular synthesis.

[7] Note that this is a sufficient albeit not necessary condition.

REFERENCES

[1] A Clark and R Eyraud, 'Polynomial identification in the limit of substitutable context-free languages', *Journal of Machine Learning Research*, (2007).

[2] F Costa and K De Grave, 'Fast neighborhood subgraph pairwise distance kernel', *Proceedings of the 26th International Conference on Machine Learning*, 255–262, (2010).

[3] Alon Halevy, Peter Norvig, and Fernando Pereira, 'The Unreasonable Effectiveness of Data', *IEEE Intelligent Systems*, **24**(2), 8–12, (2009).

[4] Zellig S. Harris, 'Distributional structure.', *Word*, (1954).

[5] M Hartenfeller and et al, 'De novo drug design', *Methods Mol Biol*, (2011).

[6] E Jeltsch and HJ Kreowski, 'Grammatical inference based on hyperedge replacement: a summary', *IEE Colloquium on Grammatical Inference: Theory, Applications and Alternatives*, 7, (1993).

[7] Jeroen Kazius, Ross McGuire, and Roberta Bursi, 'Derivation and validation of toxicophores for mutagenicity prediction', *Journal of Medicinal Chemistry*, **48**(1), 312–320, (2005).

[8] Jacek P. Kukluk, Lawrence B. Holder, and Diane J. Cook, 'Inference of edge replacement graph grammars', *International Journal on Artificial Intelligence Tools*, **17**(3), 539–554, (2008).

[9] T Oates, S Doshi, and F Huang, 'Estimating maximum likelihood parameters for stochastic context-free graph grammars', *Inductive Logic Programming*, 281–298, (2003).