The 29^{th} ILP conference, 3–5 Sep 2019, Plovdiv, Bulgaria.

Unsupervised Learning of Functional Groups for Computational Chemistry

Can Erten, Eyad Algahtani, Ian Fairlamb, Eduardo Garcia-Padilla, Jason Lynam, Suresh Manandhar, John Slattery, and Dimitar Kazakov

University of York, York, UK
dimitar.kazakov@york.ac.uk
http://www.cs.york.ac.uk/∼kazakov

Keywords: Unsupervised learning · Inductive Logic Programming · Feature generation · Concurrent computation · Computational Chemistry.

1 Introduction

Computer modelling now forms a standard part of chemists' armoury. While some models are based on a combination of precise measurements and quantum theory, others try to link what knowledge there is available about a given chemical compound with its observed properties through Machine Learning (ML) models.

Composition and structure are always the main aspects reflected in the training data, but ML approaches differ in the use of additional features signalling the presence of certain substructures that have been deemed of relevance. At one end, connectionist approaches of the 'Deep Learning' (DL) kind rely on multiple layers of neurons to combine the original aspects of the representation through non-linear functions into features of increasingly higher level of complexity and abstraction; at the other extreme, more traditional statistical ML approaches rely on the explicit provision of such features identified at an earlier stage, and added to a given data set through pre-processing.

The amount of feature construction effort saved through DL is most obvious when the native representation of the data maps easily onto the input layer of the neural network: image processing is one such class of applications. When potentially complex structure needs to be represented, as in Chemistry, labels identifying additional structural features are also explicitly added to the data. This extra effort comes on top of DL's inherent downside of needing large training data sets.

Relational, logic-based representations, such as the ones used in Inductive Logic Programming (ILP) have proven very well suited to chemistry data. Their claimed advantage is model transparency and the ability to learn from a limited number of training examples, which are annotated as a rule.

Here we are interested in what can be achieved through learning from unannotated data, with a focus on identifying functional groups, as such parts of a molecule could be used as features to assist further machine learning tasks. For

this purpose, we return to our previous work on the learning of word morphology, which showed how general learning biases related to Occam's Razor and Minimum Description Length can be used to identify functional components in unannotated lists of words. These functional components can then be used to label the training data and present it to an ILP algorithm [1, 2]. The result of such hybrid approach is twofold: a list of morpheme-like components which have the ability to recombine into new words, and a set of rules that split a given word into such functional constituents.

The bias guiding the unsupervised learning step is based on the definition of a morpheme as a functional constituent that appears next to another morpheme in at least two different words. While this definition appears circular, there is a way to square that particular circle using only four words in the process, as first shown by Saussure [3]:

$$\begin{aligned} work + s &\sim work + ing \\ sleep + s &\sim sleep + ing \end{aligned} \tag{1}$$

In this so-called *analogy* square ('works' is to 'working' what 'sleeps' is to 'sleeping'), all four constituents satisfy the above definition when considered simultaneously. Here we adopt a similar approach to the definition of functional group, and look for 4-tuples of molecules forming the same pattern, e.g.:

$$\begin{aligned} Na + OH &\sim Na + Cl \\ K + OH &\sim K + Cl \end{aligned} \tag{2}$$

Here the operator + represents a valid split for the given representation, e.g. any subset of atoms and its complement if only the classical chemical formula is considered or the result of a valence bond removal when this breaks the molecular structure into two constituents.

Given a list of molecules, one can select one, e.g. NaOH, and look for all sets of 3 molecules that complete an analogy square of the type shown above. We then: (a) assume that each split supported by at least one such square results in two valid functional groups, and, (b) note the split supported by the largest number of squares as the most likely one.

2 Sources and Representation of Data

There are several public databases containing chemical data using various representations. The two representations we have been most interested in are SMILES, and MOL3D. The former is a lossy representation and does not fully describe the three-dimensional structure of the molecule. The latter allows one to reconstruct all bonds between atoms, and this is the one we focus on, using the ChemSpider database.

3 The Method

The unsupervised part of the learning proceeds as follows. First, we consider all possible splits of a molecule into two constituents using the following simple approach: each of the bonds in the molecule is removed in turn, and an algorithm is run to detect whether all atoms remain connected, or the result is a split into two. In the latter case, the viability of this split is tested by searching for analogy squares supporting that split. If more than one split is possible, the one with the highest level of support is kept.

The search for analogy squares is computationally very expensive. We have however developed a tailored concurrent GPU-powered algorithm to speed up this computation. The algorithm takes advantage of a matrix representation of a list of molecules with each row corresponding to a molecule, and each column corresponding to a specific chemical element, e.g.:

```
      K Na O H
  KOH 1  0 1 1
 NaOH 0  1 1 1
```

Individual constituents produced by a split can be represented in the same way. The operations of considering combinations of known constituents in order to check whether they form a known molecule (i.e. one from the list) can be represented as an addition of matrices, which is easy to parallelise.

Ultimately, given a list of N molecules, the result is a list of up to N pairs of constituents generated in this way. These splits can now be used to learn ILP segmentation rules in same way as was done for word morphology [1].

Another refinement is however possible before the ILP supervised learning step is applied. One can use a simple Expectation Maximisation (EM) procedure to take into account the full list of segmented molecules in order to refine the list of splits. The EM procedure alternates between the following two steps:

1. A dictionary of all constituents appearing in the data is created, which also contains the relative frequency of each constituent, e.g. from `Na+OH, K+OH` we generate: `Na:0.25, K:0.25, OH:0.5`.
2. For each molecule, the likelihood of each possible split is estimated as the product of the relative frequencies of the two constituents, and the one with the highest likelihood is kept.

These two steps are repeated until convergence is reached.

4 Experimental Results

We have four data sets of increasing complexity and size of the order of hundreds of examples extracted by an expert from the ChemSpider database.

Fig. 1 is a sample of four molecules, where Col. 1 shows a chemical formula, Col. 2 uses the SMILES representation of the same compound, and Col.3–4 list

pairs of constituents generated by the analogy squares procedure applied to a longer list of molecules containing these four. As typical for the SMILES representation, all hydrogen atoms have been omitted. Note the two alternative splits

Table 1. Output of analogy squares procedure

Formula	SMILES	Const.1	Const.2
C3NO	CC(=O)NC	C2	CNO
C2NO	CNCO	C	CNO
C4O	C=CCC=O	C	C3O
C4O	C=CCC=O	C2	C2O
C3O	CCOC	C	C2O

for the same molecule in rows 3–4. After the application of the EM procedure, only one split is left for each molecule, while the alternative split in Row 3 is eliminated.

5 Discussion

The first results show that the idea borrowed from word morphology produces results that appear plausible on inspection by a domain expert. The parallelisation of the computationally heavy search for analogy squares makes the processing of much larger data sets potentially viable, and the ways it scales up will be tested once suitable data sets are prepared. We believe applying supervised learning to these results shows great promise, and intend to apply ILP to the task as we did with learning word constituents (and word segmentation rules) in the past.

References

1. Kazakov, D. and S. Manandhar. Unsupervised learning of word segmentation rules with genetic algorithms and inductive logic programming, *Machine Learning*, April-May 2001, vol. 43, pp. 121-162.
2. Kazakov, D., *Achievements and prospects of learning word morphology with inductive logic programming*. In James Cussens and Saso Dzeroski, editors, Learning Language in Logic, pp. 89–109. Springer, 2000. Lecture Notes in Computer Science. Vol. 1925.
3. Saussure, F. de: Cours de linguistique générale (1916)