Structured Learning Modulo Theories

Stefano Teso[1,2] Roberto Sebastiani[1] Andrea Passerini[1]

1 – Department of Information Engineering and Computer Science, University of Trento, Trento, Italy
2 – Fondazione Bruno Kessler, Trento, Italy
{teso,roberto.sebastiani,passerini}@disi.unitn.it

Methods developed in the fields of statistical relational learning (SRL) [4] and structured-output learning [1] allow to perform learning, reason and make inference about multiple relational entities characterized by both hard and soft constraints. Most such methods rely on some form of finite First-Order Logic (FOL) to encode the learning problem, and define the constraints as (weighted) logical formulae. One issue with these approaches is that FOL is not suited for efficiently reasoning over domains characterized by both *continuous* and *discrete* variables. In addition, standard FOL automated reasoning techniques offer no mechanism to deal efficiently with operators among numerical variables, like comparisons (e.g. "less-than", "equal") and arithmetical operations (e.g. summation). Many real-world domains however are inherently hybrid; this is especially true in *constructive* machine learning tasks, where the focus is on the *de-novo* design of objects with certain characteristics to be learned from examples (e.g. a recipe for a dish, with ingredients, proportions, etc.).

In order to side-step the limitations of FOL automated reasoning tools, we propose Learning Modulo Theories (LMT) [6], a max-margin structured-output learning method based on the flexibility of Satisfiability Modulo Theories (SMT) [2]. SMT languages correspond to decidable fragments of First-Order Logic, where formulae can contain Boolean variables and connectives mixed with symbols defined by a theory $\mathcal{T}$, e.g. linear arithmetic over the rationals $\mathcal{LA}(\mathbb{Q})$. For instance, the SMT($\mathcal{LA}(\mathbb{Q})$) syntax allows to write constraints such as `touching_i_j` $\leftrightarrow ((x_i + dx_i = x_j) \vee (x_j + dx_j = x_i))$ where the variables are Boolean (`touching_i_j`) or rational (x_i, x_j, dx_i,dx_j). In LMT we exploit the ability of recent solvers to solve *Optimization Modulo Theory* (OMT) problems [5], which consist in finding a model (variable assignment) that minimizes the value of some arithmetical term.

Inference in LMT fits into the classical structured-output framework, and as such predicting the best output $\boldsymbol{y}$ for any given input $\boldsymbol{x}$ boils down to maximizing the weighted compatibility function f:

$$\boldsymbol{y}^* = \mathrm{argmax}_{\boldsymbol{y}} f(\boldsymbol{x}, \boldsymbol{y}) = \mathrm{argmax}_{\boldsymbol{y}} \boldsymbol{w}^\top \boldsymbol{\psi}(\boldsymbol{x}, \boldsymbol{y})$$

Here the compatibility function is defined over a joint feature space $\boldsymbol{\psi}(\boldsymbol{x}, \boldsymbol{y})$ where the individual features are encoded in SMT, and can represent both *indicators* of (un)satisfied formulae or *costs* associated with linear arithmetical constraints. All (soft) constraints are associated with weights, which are learned from the data using the machinery of structured output Support Vector Machines [7]. Since the constraints can be encoded in SMT, both inference and learning (under a suitable loss function, such as the Hamming loss) can take advantage of very efficient OMT solvers, such as OptiMathSAT5 [3].

We applied LMT to the task of automatic character drawing, which can be framed as follows: *given a noisy image of a hand-drawn character, construct (draw) an equivalent vectorial representation.* In particular, we assume the output is a polyline made of a given number m of directed segments, each identified by a starting point (x^b, y^b) and an ending point (x^e, y^e). Intuitively, any good output $\boldsymbol{y}$ should (i) be as similar as possible to the noisy image, i.e. cover as many filled pixels as possible, and (ii) it should "look like" the corresponding vectorial character. The background knowledge includes useful predicates that express statements such as "segment i is diagonal and upwards" ($\texttt{increasing\_diagonal}(i)$), "segments i and j are connected head-to-tail" ($\texttt{h2t}(i, j)$) or compute the coverage of character pixels($\texttt{coverage} := \frac{1}{|P|} \sum_{p \in P} \mathbb{1}(\texttt{covered}(p))$). The cost function is defined as:

$$
\begin{aligned}
cost := \boldsymbol{w}^\top (\, & \underbrace{increasing(i), decreasing(i), right(i)}_{\text{for all segments } i}, \\
& \underbrace{h2t(i, i+1), t2h(i, i+1), h2h(i, i+1), t2t(i, i+1)}_{\text{for all segments } i}, coverage)
\end{aligned}
$$

so that the sequence of segment and connection types delineating the vectorial representation of a character has to be learned from data in terms of appropriate weights.

We learned a model for each of the first five letters, using 5 annotated images for training, and used it to infer the vectorial representation of the remaining 34 images[1]. The experiments were run on 8×8 downscaled images for efficiency. We report a sample output generated by LMT in Figure 1. The experiments show that, despite the very noisy annotations, LMT is indeed able to address the character drawing problem and produce reasonable outputs for all target letters.

[1] Dataset taken from http://cs.nyu.edu/∼roweis/data.html

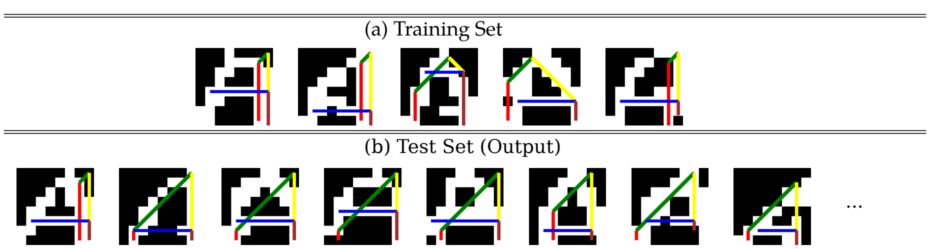

Fig. 1. Sample taken from the character drawing experiments, see [6] for the complete details.

References

1. Bakir, G.H., Hofmann, T., Schölkopf, B., Smola, A.J., Taskar, B., Vishwanathan, S.V.N.: Predicting Structured Data (Neural Information Processing). The MIT Press (2007)
2. Barrett, C., Sebastiani, R., Seshia, S.A., Tinelli, C.: Satisfiability Modulo Theories, chap. 26, pp. 825–885. Frontiers in Artificial Intelligence and Applications, IOS Press (February 2009)
3. Cimatti, A., Griggio, A., Schaafsma, B.J., Sebastiani, R.: The MathSAT 5 SMT Solver. In: Tools and Algorithms for the Construction and Analysis of Systems, TACAS'13. LNCS, vol. 7795, pp. 95–109. Springer (2013)
4. Getoor, L., Taskar, B.: Introduction to statistical relational learning. The MIT press (2007)
5. Sebastiani, R., Tomasi, S.: Optimization in SMT with LA(Q) Cost Functions. In: proc. International Joint Conference on Automated Reasoning, IJCAR. LNAI, vol. 7364, pp. 484–498. Springer (July 2012)
6. Teso, S., Sebastiani, R., Passerini, A.: Structured learning modulo theories. arXiv preprint arXiv:1405.1675 (2014)
7. Tsochantaridis, I., Joachims, T., Hofmann, T., Altun, Y.: Large margin methods for structured and interdependent output variables. J. Mach. Learn. Res. 6, 1453–1484 (Dec 2005), `http://dl.acm.org/citation.cfm?id=1046920.1088722`