Introduction to the special issue on Combining Constraint Solving with Mining and Learning

Andrea Passerini

andrea.passerini@unitn.it
DISI, University of Trento, Italy

Guido Tack

guido.tack@monash.edu
Faculty of Information Technology, Monash University, Australia

Tias Guns

tias.guns@cs.kuleuven.be and tias.guns@vub.ac.be
Dept. Computer Science, KU Leuven, Belgium, and
Dept. Business, Technology and Operations, VUB, Belgium

1. Introduction

Data mining, machine learning and constraint solving are major themes in artificial intelligence research. They have evolved quite independently, though in recent years, there is a growing interest in the potential of integrating these fields.

Data mining and machine learning are methods for extracting regularities out of data, for example which instances cluster together, what patterns appear frequently in the data or what function can discriminate positive from negative examples. On the other hand, constraint solving investigates generic methods for solving constraint satisfaction and optimization problems.

Until now, these fields have evolved quite independently. Nevertheless, complex mining and learning tasks can often be formalized in terms of constraints that need to be satisfied, ranging from logical constraints to (non-linear) numeric ones. This methodology is becoming increasingly compelling, as the community is progressively moving from addressing relatively simple tasks on tabular data to complex problems on structured data. The use of machine learning and data mining techniques to enhance constraint solving is equally promising, and only starting to be explored. Consequently, there is a mutual benefit in the study of these approaches, which can lead to advances in both fields.

For this reason, a number of workshops have been organized on the topic, as well as this special issue. A combination, or integration, of Machine Learning (ML) and Data Mining (DM) with Constraint Solving (CS) can work in two

directions: CS techniques can be used to (partly) solve tasks in DM/ML; Alternatively, one can use DM/ML techniques to augment CS techniques. Both directions are increasingly being explored. Yet, a proper approach in either direction requires good knowledge of both research fields, and hence feedback from both communities. The aim of this special issue is to bundle recent advances on combinations of the two fields, and to foster interaction between researchers of both communities.

The integration of machine learning and constraint solving is rooted in recent trends in these research areas. Fields like statistical relational learning [1], probabilistic programming [2] and structured-output prediction [3] naturally integrate learning algorithms with reasoning engines for addressing the underlying sets of constraints. Reasoning engines on their own, including SAT solvers [4], constraint solvers [5], mixed integer programming [6] as well as high-level modeling languagues such as OPL [7] and MiniZinc [8] are extending both the scale and diversity of problems that can be addressed, as well as making the technology more accessible to other researchers.

A number of developments in recent years have extended the reach and applicability of integrations between the two fields. In particular: an increasing number of discrete and symbolic data mining and learning problems, for which other mathematical programming approaches are less suited; the rise of interest in 'constraint-based' mining and learning and the applicability of generic constraint solvers for this purpose; the interest in learning or predicting structured output, that is objects and their relations rather than a class label. In general, the increased presence of hard constraints in machine learning and data mining creates a need for generic and flexible methods that deal with such hard constraints. This need, combined with the advancement of both the speed of computer hardware as well as the scalability and maturity of constraint solving systems, is creating new opportunities for interaction between these methods.

There is also an increasing interest in using machine learning to improve the solving of constraint problems, as well as improving the modeling and the embedding of learned constraints and objectives into the model. This is driven in part also by advances in computer hardware and the availability of large storage capacities and multiple processing units, but especially by the increased availability of data regarding all aspects of the solving process, such as: internal data regarding the solving and the effect of solver parameters that can be used to learn and automatically adapt the solving behavior; the use of external data regarding the problem domain that can be used to learn the parameters of constraints or even entire parts of the constraint model; and the use of external data to infer objective functions or learn other functions that could not otherwise be formalized in terms of simple constraints. These integrations are non-trivial, and many research questions remain regarding how to best do this.

In the following we briefly introduce the papers in this special issue, grouping them according to the aspects of integration they focus on.

2. Machine learning and data mining with constraints

In today's data-rich world, machine learning and data mining techniques allow us to extract knowledge from data. However, such knowledge can take many forms and often depends on the application and operational context at hand. In many cases, one will wish to guide the learning and mining process by adding information on the type of knowledge we are seeking. One common way to express this is through constraints.

In recent years, constraint solving has been shown to offer a generic methodology that fits many mining and learning settings. This special issue contains recent advancements in the following broad research domains: constraint solving in pattern mining, clustering and learning.

2.1. Pattern mining using constraint solving

Pattern mining and constraint-based mining seeks to extract patterns from a large set of observations that satisfy the provided constraints, for example that a pattern must appear in a given number of observations. Using constraint solving techniques allows for a generic technique that can express a wide range of constraints and tasks. The following papers investigate the possibilities for different classes of tasks.

MiningZinc: a declarative framework for constraint-based mining *Tias Guns, Anton Dries, Siegfried Nijssen, Guido Tack and Luc De Raedt*

The authors aim at bridging the gap between problem-specific pattern mining procedures and declarative problem solving methods. They present MiningZinc, a modeling language allowing to specify data mining tasks as constraint solving problems. The language is coupled with an execution mechanism which automatically extracts different execution strategies combining generic and specialized solvers. Tasks ranging from closed itemset mining to discriminative pattern mining and mining pattern sets can be modelled and solved in this framework.

Mining Top-k Motifs with a SAT-based Framework *Said Jabbour, Lakhdar Sais and Yakoub Salhi*

This work introduces a generalization of partial max-SAT and computing X-minimal models called Top-k SAT. Given a preference relation, it finds those solutions that have fewer than k other solutions preferred to it. This framework and the proposed algorithm is then for constrained itemset mining and sequence mining. The authors show how it can be used to mine top-k closed itemsets under size constraints, as well as top-k sequences of items and sequences of itemsets.

Skypattern Mining: from Pattern Condensed Representations to Dynamic Constraint Satisfaction Problem *Willy Ugarte, Patrice Boizumault, Bruno Cremilleux, Alban Lepailleur, Samir Loudni, Marc Plantevit, Chedy Raissi and Arnaud Soulet*

The use of a preference relation is studied in this work from a different angle: the paper studies the task of finding all Pareto-optimal solutions given a set of measures. It investigates under what conditions the number of measures can be safely reduced, through *skylineability*. It continues to compare and unify a traditional approach of post-processing the output of a specialized method with the use of a generic constraint solver and shows the strengths and weaknesses of each.

2.2. Clustering using constraint solving

In clustering the goal is to group instances based on similarity. The following works in the special issue demonstrate that constraint solving is applicable to many different forms of clustering:

Constrained Clustering by Constraint Programming *Thi-Bich-Hanh Dao, Khanh-Chuong Duong and Christel Vrain*

A main body of work is on clustering points into a partition. This work presents a generic partition-based clustering framework using constraint programming, which finds exact, optimal clusters. Multiple clustering measures as well as clustering constraints can be expressed and combined in this framework. It is also shown how it can be used for multi-objective clustering.

A Flexible ILP Formulation for Hierarchical Clustering *Sean Gilpin and Ian Davidson*

A widely used clustering algorithm in practice is hierarchical clustering, which returns a *dendogram* or hierarchy instead of a partitioning. As the authors show this problem can be expressed as an Integer Linear Programming problem. This also allows for expressing additional constraints on the instances and the structure, as well as relaxing some constraints to find overlapping groups.

Cost-Optimal Constrained Correlation Clustering via weighted Partial Maximum Satisfiability *Jeremias Berg and Matti Jarvisalo*

Correlation clustering assumes that each pair of instances can be classified as being similar or not similar, and the goal is to find a clustering that correlates well with this classification. The authors show how this problem can be transformed into a (weighted) partial maxSAT problem, and hence such solvers can be used to find the optimal solution efficiently. They show how a number of other constraints can be incorporated as well.

2.3. Learning using constraint solving

In the following we summarize the contributions which incorporate constraint solving approaches into inference and learning algorithms.

Semantic-Based Regularization for Learning and Inference *Michelangelo Diligenti, Marco Gori and Claudio Saccà.*

Semantic-based regularization is a framework for statistical relational learning combining first order logic and kernel machines. The idea is that of using kernel

machines as base statistical learners for individual target predicates, and first order logic to introduce constraints enforcing known relationships between predicates. The authors provide a broad overview of the framework, its relationship with existing paradigms, and extend it in a number of directions.

Structured Learning Modulo Theories *Stefano Teso, Roberto Sebastiani and Andrea Passerini* The paper introduces a new way for both learning from structured input as well as predicting structured output. The approach is centered around the definition of hard constraints and soft constraints over Boolean and real variables. Structured SVM learning can then be used to learn the weights of the soft constraints, where SAT module optimization is used for both the SVM learning its separation problem as well as for inferring the best prediction for a given input.

Relational Linear Programming *Kristian Kersting, Martin Mladenov and Pavel Tokmakov*
The authors introduce a framework combining linear and logic programs, two forms of constraint solving. The framework allows to express linear programs involving varying numbers of variables and constraints in a relational fashion, as templates to be compiled into ground logic programs when applied to a knowledge base. The authors additionally developed a lifted linear programming approach based on fractional automorphisms to speed up the optimization process. They show the potential of the proposed approach on a range of machine learning and artificial intelligence tasks.

Learning an efficient constructive sampler for graphs *Fabrizio Costa*
Learning to generate novel structures from a set of examples is an extremely challenging task, with a wide range of potential applications. The author presents a Metropolis-Hastings sampler for graph generation combining local moves in the space of graphs with probability estimation of the quality of the resulting candidate. Local moves are generated by upgrading the *substitutability* principle from strings to graphs, for which validity of a change is based on compatibility of its context. The method is shown to generate highly structured and constrained graphs of good predictive quality.

Learning Bayesian Networks under Equivalence Constraints *Tiansheng Yao, Arthur Choi and Adnan Darwiche*
Given a partially observed dataset, an equivalence constraint for a variable is a set of instances sharing the same (unknown) value for that variable. The resulting constrained log-likelihood is intractable even for networks where standard inference is tractable. The authors propose an efficient approximate algorithm and prove its effectiveness on a number of experimental scenarios. They also show how the framework allows to naturally address tasks like semi-supervised clustering and topic model refinement.

Integer Linear Programming for the Bayesian Network Structure

Learning Problem *Mark Bartlett and James Cussens*
This paper discusses ILP techniques for learning the structure of Bayesian Networks. The problem is addressed by a branch-and-cut procedure, where branching is conducted on variable values and linear relaxations of the resulting ILP subproblems are iteratively solved. Cutting planes are added during the search to prune away as much of the search space as possible without removing relevant candidate solutions. The authors present an in-depth analysis of various types of cuts and their influence on the speed of convergence of the algorithm.

3. Constraint solving using machine learning

The close connection between constraint solving and machine learning is exemplified not only by the impact that constraint solving has had on machine learning and data mining, as discussed above, but also by the opposite direction.

In this special issue three such areas are explored where machine learning techniques have been applied to improve constraint solving: algorithm selection, constraint acquisition, and optimization over learned models.

3.1. Learning for constraint solver and parameter selection

The goal here is to collect data about the empirical performance of different algorithms and their configurations for different problem instances, and use that to learn when to use what.

Automatic Construction of Parallel Portfolios via Algorithm Configuration *Marius Lindauer, Holger Hoos, Kevin Leyton-Brown and Torsten Schaub*
This article applies automatic algorithm selection and configuration techniques to the problem of automatically constructing *parallel* SAT solver portfolios from one or multiple, sequential or parallel, basic SAT algorithms. The approach has been evaluated on standard SAT benchmarks, and the results show that this method can produce portfolios that outperform any sequential portfolio solver, without requiring complex parallel code to be developed.

Algorithm Recommender System. Application to Algorithm Portfolio Selection *Mustafa Misir and Michele Sebag*
The authors describe Alors, a novel recommender system for algorithm selection. The system combines collaborative filtering approaches for sparse matrix completion with learning of latent representations for cold-start recommendation on novel problem instances. A deep and extensive evaluation highlights the advantages of the proposed approach in learning from sparse evidence, recommending algorithms for completely novel problem instances and identifying potential reasons in case of suboptimal recommendations.

3.2. Learning constraints

Constraint Acquisition *Christian Bessiere, Frdric Koriche, Nadjib Lazaar and Barry O'Sullivan*

In this article, the authors tackle the difficult problem of helping non-expert users of constraint programming tools to model their problems. The approach presented here is to learn constraint networks from positive and negative examples, as classified by the user. This article introduces the basic definitions of the *constraint acquisition* problem, and then presents several variants of the Conacq system, which uses version space learning to acquire constraint networks based on user feedback.

3.3. Optimization over learned models

When dealing with real-world systems, it is not always possible to model the problem or system explicitly in terms of constraints. This can be due to the inherent complexity of the system, e.g. physical systems, or it can be due to inherent dynamics, where the system changes over time.

An alternative approach is to *observe* the system and *learn* the underlying relations using machine learning techniques. By integrating this learned model into the optimization model, one can use it as part of the constraint specification.

Empirical Decision Model Learning *Michele Lombardi, Michela Milano and Andrea Bartolini*

This paper investigates this approach of integrating a learned model into an optimization model for a thermal-aware dispatching problem on a multi-core CPU, and dubs it Empirical Model Learning (EML). On the machine learning side, neural networks and decision trees are used and it is shown how the resulting models can be expressed in multiple constraint solving formalisms, namely Local Search, Constraint Programming, Mixed Integer Non-Linear Programming and SAT Modulo Theories.

Auction Optimization using Regression Trees and Linear Models as Integer Programs *Sicco Verwer, Yingqian Zhang and Qing Chuan Ye*

The order in which items are auctioned can influence the profit of the auction. This paper proposes to use machine learning to predict the expected profit of an auction order. Furthermore, it is shown how learned linear regression models and regression trees can be modeled in Integer Linear Programming. Together, the auction ordering can be optimized using the learned model its output as objective function.

4. Acknowledgment

We would like to thank the co-organizers of the different CoCoMile workshops Barry O'Sullivan, Remi Coletta and Lars Kotthoff. We would also like to thank Luc De Raedt and Siegfried Nijssen for their support.

References

[1] L. Getoor, B. Taskar, Introduction to Statistical Relational Learning, The MIT Press, 2007.

[2] A. D. Gordon, T. A. Henzinger, A. V. Nori, S. K. Rajamani, Probabilistic programming, in: International Conference on Software Engineering (ICSE Future of Software Engineering), IEEE, 2014.

[3] G. H. Bakir, T. Hofmann, B. Schölkopf, A. J. Smola, B. Taskar, S. V. N. Vishwanathan, Predicting Structured Data, The MIT Press, 2007.

[4] A. Biere, M. J. H. Heule, H. van Maaren, T. Walsh (Eds.), Handbook of Satisfiability, Vol. 185 of Frontiers in Artificial Intelligence and Applications, IOS Press, 2009.

[5] F. Rossi, P. van Beek, T. Walsh, Handbook of Constraint Programming (Foundations of Artificial Intelligence), Elsevier Science Inc., 2006.

[6] G. L. Nemhauser, L. A. Wolsey, Integer programming and combinatorial optimization, Wiley, Chichester. GL Nemhauser, MWP Savelsbergh, GS Sigismondi (1992). Constraint Classification for Mixed Integer Programming Formulations. COAL Bulletin 20 (1988) 8–12.

[7] P. Van Hentenryck, The OPL optimization programming language, MIT Press, 1999.

[8] N. Nethercote, P. J. Stuckey, R. Becket, S. Brand, G. J. Duck, G. Tack, Minizinc: Towards a standard CP modelling language, in: Principles and Practice of Constraint Programming, Vol. 4741 of Lecture Notes in Computer Science, Springer, 2007, pp. 529–543.