Report from Dagstuhl Seminar 14411

# Constraints, Optimization and Data

**Edited by**

**Luc De Raedt[1], Siegfried Nijssen[1,2], Barry O'Sullivan[3], and Michele Sebag[4]**

1 **KU Leuven, BE,** firstname.lastname@cs.kuleuven.be
2 **Universiteit Leiden, NL,** s.nijssen@liacs.leidenuniv.nl
3 **University College Cork, IE,** b.osullivan@cs.ucc.ie
4 **Université Paris Sud, FR,** michele.sebag@lri.fr

## Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 14411 "Constraints, Optimization and Data". Constraint programming and optimization have recently received considerable attention from the fields of machine learning and data mining; similarly, machine learning and data mining have received considerable attention from the fields of constraint programming and optimization. The goal of the seminar was to showcase recent progress in these different areas, with the objective of working towards a common basis of understanding, which should help to facilitate future synergies.

**Seminar** October 6–10, 2014 – http://www.dagstuhl.de/14411
**1998 ACM Subject Classification** H.2.8 Database applications. F.4.1. Mathematical logic
**Keywords and phrases** Data mining, constraint programming, machine learning
**Digital Object Identifier** 10.4230/DagRep.4.10.1

## 1 Executive Summary

*Luc De Raedt*
*Siegfried Nijssen*
*Barry O'Sullivan*
*Michele Sebag*

Constraint programming and optimization (CPO) have recently received considerable attention from the fields of machine learning and data mining (MLDM). On the one hand, the hypotheses and patterns that one seeks to discover in MLDM can be specified in terms of constraints (e. g. labels in the case of supervised learning, preferences in the case of learning to rank, must-link and cannot-link in the case of unsupervised learning, coverage and lift in the case of data mining). On the other hand, powerful constraint programming solvers have been developed. If MLDM users express their requirements in terms of constraints they can delegate the MLDM process to such highly efficient solvers.

Conversely, CPO can benefit from integrating learning and mining functionalities in a number of ways. For example, formulating a real-world problem in terms of constraints requires significant expertise in the problem domain. Also, selecting the most appropriate constraints, in terms of constraint solving efficiency, requires considerable expertise in the CPO domain. In other words, experience plays a major role in successfully applying CPO technology.

In addition, both CPO and MLDM share a common challenge associated with tuning their respective methods, specifically determining the best parameters to chose for an algorithm depending on the task at hand. A typical performance metric in machine learning is the predictive accuracy of a hypotheses, while in CPO it might be search cost or solution quality.

This seminar built upon the 2011 *Constraint Programming meets Machine Learning and Data Mining*[1] and the 2014 *Preference learning*[2] seminars. Its goal was to identify the key challenges and opportunities at the crossroads of CPO and MLDM. The interests of the participants included the following:

- Problem formulation and modelling: constraint-based modelling; preference formalisms; loss functions in ML; modelling and exploiting background knowledge; structured properties (e. g. preserving spatio-temporal structures).
- Improvement of algorithms / platforms in the areas of algorithm selection, algorithm configuration, and/or algorithm scheduling, particularly with respect to parallel execution.
- Specification and reasoning about goals and optimization criteria: modelling preferences and integrating with human expertise (exploiting the "human in the loop") to converge on high quality outcomes.
- Additional functionalities such as the use of visualization and explanation.
- Algorithmic scalability.
- Approximate reasoning, reasoning under uncertainty, and incorporating probability.

The seminar was organized into seven sessions: frameworks and languages; algorithm configuration; constraints in pattern mining; learning constraints; machine learning with constraints; applications; and demonstrations. The demonstrations presented at the seminar were by:

- Guido Tack – MiniZinc (see http://mininzinc.org);
- Joaquin Vanschoren – OpenML (see http://openml.org);
- Tias Guns – MiningZinc (see http://dtai.cs.kuleuven.be/CP4IM/miningzinc);
- Bruno Crémilleux – software for the calculation of Sky Pattern Cubes;
- Marc Denecker – IDP (see http://dtai.cs.kuleuven.be/krr/software/idp);
- Holger Hoos – algorithm selection and portfolio software;
- Luc De Raedt – ProbLog (see http://dtai.cs.kuleuven.be/problog/).

The seminar also had five working groups on:

- Declarative Languages for Machine Learning and Data Mining;
- Learning and Optimization with the Human in the Loop;
- Meta-Algorithmic Techniques;
- Big Data;
- Towards Killer Applications.

[1] http://www.dagstuhl.de/11201
[2] http://www.dagstuhl.de/14101

## 2 Table of Contents

## 3 Overview of Talks

### 3.1 The road to declarative data analysis is paved with constraints

*Hendrik Blockeel (KU Leuven, BE)*

Data analysis is declarative when the user can just formulate the question to be answered, rather than the method for solving that question. This talk presented arguments and illustrations for the following claims: (1) Declarative approaches will make data analysis easier, more flexible, more efficient and more correct. (2) Constraint-based reasoning will play a major part in reaching this objective.

### 3.2 Constraint-based mining: preliminary results on deriving constraints from experts models

*Jean-François Boulicaut (INSA – Lyon, FR)*

**Joint work of** Boulicaut, Jean-François; Flouvat, Fréderic; Selmaoui, Nazha; Sanhes, Jérémy
**URL** http://liris.cnrs.fr/~jboulica/

First, we discuss the basics of pattern domain design (specifying pattern languages over specific types of data, introducing measures and more or less related primitive constraints and the way to combine them as well). In fact, thanks to various recent work in our team (e. g., mining preserved cliques in relational dynamic graphs [1], gradual pattern mining in attributed graphs [2], mining sets of cohesive nodes in dynamic attribute graphs [3]), we can easily refer to the different roles of primitive constraints when considering one inductive query: some express pattern semantics, others are dedicated to pattern objective interestingness while the remaining ones specify subjective interestingness issues. We can then consider both the declarative and the computational views on the specified mining tasks. It is important to address the relationships of primitive constraint properties w.r.t. typical enumeration strategies over the pattern language. Notice that the terminology about constraint-based data mining and inductive database topics can be found in the following books [4, 5, 6].

In a second step, we report about recent results that we obtained thanks to a cooperation between INSA de Lyon and the University of New Caledonia. We consider the context where expert models are available as multivariate real functions. Then, we study how to derive a new type of primitive constraint that can exploit such a model by specifying that the model may predict a value higher than a user-defined threshold for the interesting patterns. As a concrete example, we introduce a case study on soil erosion in New Caledonia. We set up a simple item set mining context where transactions correspond to areas (more precisely pixels in satellite images) and items corresponds to various properties of the areas. Using the new constraint derived from models that evaluate an erosion risk, we can enforce the data mining tasks to focus on itemsets that are associated to areas for which the model predict a high enough risk. Studying the properties of the new primitive constraint and finding pruning rules that can be easily combined with the other pruning rules for item set mining is the main technical contribution. Perspectives are obvious like deriving new primitive constraints from the same kind of model, looking for new king of models, looking for more sophisticated data mining tasks (than just item set mining). This preliminary work has been published in [7].

**References**

1 Loïc Cerf, Bao Nhan Nguyen Tran, Jean-Francois Boulicaut. Mining constrained cross-graph cliques in dynamic networks. In: Inductive Databases and Constraint-Based Data Mining, S. Dzeroski, B. Goethals, and P. Panov (Eds), pp. 199–228, November 2010, Springer.

2 Adriana Prado, Marc Plantevit, Céline Robardet, Jean-Francois Boulicaut. Mining graph topological patterns: Finding co-variations among vertex descriptors. IEEE Transactions on Data and Knowledge Engineering 25(9):2090–2104, 2013, IEEE Computer Press.

3 Elise Desmier, Marc Plantevit, Céline Robardet, Jean-Francois Boulicaut. Trend mining in dynamic attributed graphs. Proc. European Conf. on Machine Learning and Principles and Practice of Knowledge Discovery in Databases ECML PKDD 2013, Praha, Czech Republic, September 2013. H. Blockeel, K. Kersting, S. Nijssen, and F. Zelezný (Eds.). Springer LNAI 8188, pp. 654–669.

4 Database support for Data Mining Applications – Discovering Knowledge with Inductive Queries LNCS 2682, R. Meo, P-L. Lanzi, and M. Klemettinen (Eds.), 2004. Springer.

5 Constraint-based Mining and Inductive Databases, J-F. Boulicaut, L. De Raedt, H. Mannila (Eds.) Springer LNCS 3848, 2005.

6 Inductive Databases and Constraint-Based Data Mining, S. Dzeroski, B. Goethals, and P. Panov (Eds), November 2010, Springer.

7 Frédéric Flouvat, Jérémy Sanhes, Claude Pasquier, Nazha Selmaoui-Folcher, Jean-Francois Boulicaut. Improving pattern discovery relevancy by deriving constraints from expert models. Proc. European Conference on Artificial Intelligence ECAI 2014. August 2014, Praha, Czech Republic, T. Schaub et al. (Eds), regular paper, pp. 327–332, IOS.

## 3.3 Data, learning and optimisation in home energy management

*Ken Brown (University College Cork, IE)*

**Joint work of** Brown, Ken; Manzano, Oscar; Murphy, Sean Og; Trabelsi, Walid; O'Toole, Liam; O'Sullivan, Barry

The Authentic project is developing an integrated hardware and software solution of home energy management, to provide personal tailored feedback and guidance to home users. Data on appliance use, occupancy, environmental conditions and energy consumption are gathered by sensors. We then learn models of occupant behaviour and house response, identifying areas of energy inefficiency. Finally, we develop optimisation models to guide occupants towards achieving their own energy goals. The models attempt to find the minimal change to observed behaviour which achieves the goals. The models are learned online through interaction with the occupants.

## 3.4 On Preference-based (soft) pattern sets

*Bruno Crémilleux (Caen University, FR)*

**Joint work of** Boizumault, Patrice; Crémilleux, Bruno; Loudni, Samir; Ugarte Rojas, Willy
**Main reference** W. Ugarte, P. Boizumault, S. Loudni, B. Crémilleux, "Computing skypattern cubes," in Proc. of the 21st European Conf. on Artificial Intelligence (ECAI'14), pp. 903–908, IOS Press, 2014.
**URL** http://dx.doi.org/10.3233/978-1-61499-419-0-903

In the last decade, the pattern mining community has witnessed a sharp shift from efficiency-based approaches to methods which can extract more meaningful patterns. Recently, new methods adapting results from multi criteria decision analyses such as Pareto efficiency, or skylines, have been studied. Within pattern mining, this novel line of research allows the easy expression of preferences according to a dominance relation on a set of measures and avoids the well-known threshold issue.

In this talk, we present the discovery of soft skyline patterns (or soft skypatterns) based on theoretical relationships with condensed representations of patterns and the dynamic constraint satisfaction problems framework. To avoid an apriori choice of measures, we propose to use the skypattern cube according to the set of measures. We show how to efficiently build the skypattern cube and provide a concise representation of the cube based on skypattern equivalence classes. Navigation trough the cube indicates differences and similarities between skypattern sets when a measure is added or removed, it highlights the role of the measures and helps to discover the most interesting skypattern sets. We set our work in the global picture of the discovery of pattern sets and k-pattern sets/n-ary patterns.

### References

1 Willy Ugarte and Patrice Boizumault and Samir Loudni and Bruno Crémilleux and Alban Lepailleur Mining (soft-) skypatterns using dynamic CSP. *In* 11th Int. Conf. on Integration of Artificial Intelligence (AI) and Operations Research (OR) Techniques in Constraint Programming (CP) (CPAIOR 2014), 19–23 May, 2014, Cork, Ireland, pages 71–87.
2 Willy Ugarte and Patrice Boizumault and Samir Loudni and Bruno Crémilleux Computing skypattern cubes. *In* 21st European Conference on Artificial Intelligence (ECAI 2014), 18–22 August 2014, Prague, Czech Republic, pp. 903–908. IOS Press.
3 Willy Ugarte and Patrice Boizumault and Samir Loudni and Bruno Crémilleux Computing skypattern cubes using relaxation. *In* 26th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2014), 10–12 November 2014, Limassol, Cyprus.

## 3.5 GOBNILP: CIP for BN learning

*James Cussens (University of York, GB)*

**URL** http://www.cs.york.ac.uk/aig/sw/gobnilp

In this talk I will describe the Bayesian network learning system GOBNILP [1, 2] and describe more general issues concerned with using (constraint) integer programming (CIP) for machine learning tasks. A CIP allows one to describe a machine learning task declaratively and then have a solver solve the resulting problem. However, doing so *efficiently* is not so simple, and in our experience requires detailed study and considerable implementation effort. I will discuss the lessons we have learned in this work. Chief amongst these is that with a

CIP/MIP approach constructing a tight linear relaxation is crucial. To do this it is necessary to *avoid* compact representations of the problem.

**References**

1 James Cussens. Bayesian network learning with cutting planes. In Fabio G. Cozman and Avi Pfeffer, editors, *Proceedings of the 27th Conference on Uncertainty in Artificial Intelligence (UAI 2011)*, pp. 153–160, Barcelona, 2011. AUAI Press.

2 Mark Barlett and James Cussens. Advances in Bayesian network learning using integer programming. In Ann Nicholson and Padhraic Smyth, editors, *Proceedings of the 29th Conference on Uncertainty in Artificial Intelligence (UAI 2013)*, pp. 182–191, Bellevue, 2013. AUAI Press

## 3.6 Constrained Clustering by Constraint Programming

*Thi-Bich-Hanh Dao (University of Orleans, FR)*

**Joint work of** Dao, Thi-Bich-Hanh; Duong, Khanh-Chuong; Vrain, Christel
**Main reference** T.-B.-H. Dao, K.-C. Duong, C. Vrain, "Constrained Clustering by Constraint Programming," to appear.

Constrained Clustering has received much attention this last decade. It allows to make the clustering task either easier or more accurate by integrating user-constraints. Several kinds of constraints can be considered; they may be requirements on the clusters, as for instance their sizes or their diameters, or on pairs of instances the expert knows that they must be or cannot be in the same cluster (must-link or cannot-link constraints). Much work has focused on instance-based constraints and has adapted classical clustering methods to handle such kinds of constraints, but usually considering only one optimization criterion. Few works consider different kinds of constraints or integrate different optimization criteria. In [1] a SAT based framework for constrained clustering has been proposed, but it is limited to problems with two classes. A framework based on Integer Linear Programming has also been proposed in [2] but it is more suited to conceptual clustering. Another Integer Linear Programming framework has been proposed in [3], which considers must-link, cannot-link constraints and anti-monotone constraints, but with a single optimization criterion that is the minimum sum of squares.

In this talk, we present a framework based on Constraint Programming for Constrained Clustering [4, 5]. The framework is general and declarative, it allows to choose among several optimization criteria and to integrate different kinds of user-constraints. The optimization criteria proposed are minimizing the maximal diameter of the clusters, maximizing the minimal split between clusters or minimizing the within-cluster sum of dissimilarities. The framework integrates must-link or cannot-link constraints and all popular cluster level constraints. The model is flexible since it does not require to set the number of clusters beforehand, only a lower and an upper bound on the number of clusters have to be given. Relying on the dissimilarity between objects, the model can handle quantitative or qualitative datasets, as soon as such a measure can be defined.

The approach we propose can be easily embedded in a more general process for Constrained Clustering. Considering Data Mining as an iterative and interactive process composed of the classical steps of task formulation, data preparation, choice of a learning tool thus requiring to set parameters and validation of the results, a user can specify the task at hand including or not some constraints and decide to change the settings according to the results. He/she may decide to change the constraints, removing some constraints, adding or hardening other ones. The modularity and the declarativity of our model allow this easily. In this talk, we illustrate this point on a bi-criterion clustering problem. We show that our framework can be used to find the Pareto front of a bi-criterion maximizing split-minimizing diameter problem, under consideration of user-constraints.

**References**

1 I. Davidson, S.S. Ravi, L. Shamis. *A SAT-base Framework for Efficicient Constrained Clustering*. In Proceedings of the 10th SIAM International Conference on Data Mining, pp. 94–105, 2010.

2 M. Mueller, S. Kramer. *Integer Linear Programming Models for Constrained Clustering*. in Proceedings of the 13th International Conference on Discovery Science, pp. 159–173, 2010.

3 B. Babaki, T. Guns, S. Nijssen. *Constrained Clustering using Column Generation*. In Proceedings of the 11st International Conference on Integration of AI and OR Techniques in Constraint Programming, pp. 438–454, 2014.

4 T.-B.-H. Dao, K.-C. Duong, C. Vrain. *A Declarative Framework for Constrained Clustering*. In Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, pp. 419–434, 2013.

5 T.-B.-H. Dao, K.-C. Duong, C. Vrain. *Constrained Clustering by Constraint Programming*. Artificial Intelligence Journal, accepted with minor revisions.

## 3.7 Human in the Loop Learning with Constraints

*Ian Davidson (University of California – Davis, US)*

We show how constraints can be used for active and transfer learning to facilitate human in the loop learning. The constraints are used as a mechanism for the human to have a dialog with the algorithm. The human specifies guidance to the machine in the form of constraints and the algorithm asks questions of the human also in the form of constraints. We briefly overview applications in fMRI data analysis.

## 3.8 Languages for Mining and Learning

*Luc De Raedt (KU Leuven, Belgium)*

Applying machine learning and data mining to novel applications is cumbersome. This observation is the prime motivation for the interest in languages for learning and mining. This note provides a gentle introduction to three types of languages that support machine learning and data mining: inductive query languages, which extend database query languages with primitives for mining and learning, modelling languages, which allow to declaratively specify and solve mining and learning problems, and programming languages, that support the learning of functions and subroutines. It uses an example of each type of language to introduce the underlying ideas and puts them into a common perspective. This then forms the basis for a short analysis of the state-of-the-art.

**References**

1 Luc De Raedt. *Languages for Mining and Learning*. Proc. 29th AAAI Conference on Artificial Intelligence, Senior Member Track, 2015, in press.

## 3.9 Constraint solving with extensions of classical logic Applications to data mining

*Marc Denecker (KU Leuven, BE)*

I will present a motivation for the use of (extensions of) first order logic for constraint solving. I will situate this in the context of constraint programming and answer set programming. Examples will serve to present the paradigm and some useful language extensions. I will give a brief overview of the system IDP3 and discuss implementation techniques that have been used in it and results of experiments. I can discuss the relationship with Zinc, and illustrate applications of the system to some datamining applications.

## 3.10 Learning to Build Effective Constraint Models

*Alan Frisch (University of York, GB)*

**Joint work of** Frisch, Alan; Miguel, Ian; Jefferson, Chris; Agkun, Ozgur; Martinez-Hernandez, Bernadette
**Main reference** O. Akun, I. Miguel, C. Jefferson, A. M. Frisch, B. Hnich, "Extensible Automated Constraint Modelling," in Proc. of the 25th AAAI Conf. on Artificial Intelligence (AAAI'11), pp. 4–11, AAAI, 2011; pre-print available from author's webpage.
**URL** http://www.aaai.org/ocs/index.php/AAAI/AAAI11/paper/view/3687
**URL** http://www.cs.york.ac.uk/aig/constraints/AutoModel/conjure-aaai11.pdf

CONJURE is a rule-based system that can automatically generate a set of alternative constraint models for a problem expressed in the ESSENCE problem specification language. CONJURE currently has no highly effective method to select a good model from among a set of alternatives it generates.

After a brief introduction to ESSENCE and CONJURE, this talk argues that it is better to drive the learning of model selection not by features of the models but rather by features of the design decisions made in generating the model.

**References**

1 Alan M. Frisch, Warwick Harvey, Chris Jefferson, Bernadette Martinez Hernandez and Ian Miguel. ESSENCE: A Constraint Language for Specifying Combinatorial Problems. In: *Constraints*, 13(3), July 2008.

2 Ozgur Akgun, Ian Miguel, Chris Jefferson, Alan M Frisch and Brahim Hnich. Extensible Automated Constraint Modelling. In: *Proc. of the 25th AAAI Conference on Artificial Intelligence*, 2011.

## 3.11 Classes of constraints within a high level model of constraint optimization

*Randy Goebel (University of Alberta, CA)*

The traditional logic programming appeal of all computation falling within logical semantics extends to the architecture of constraint optimization, where constraint specifications of a model, an objective function, and method constraints can be uniformly specified under a single semantics. We sketch a simple uniform architecture which helps identify how incremental dynamic constraint optimization systems can be incrementally improved. The architecture shows the fit for where probability, continuous variables, hypothetical reasoning, and belief revision.

## 3.12 MiningZinc, a declarative framework for constraint-based mining

*Tias Guns (KU Leuven, BE)*

**Joint work of** Guns, Tias; Dries, Anton; Nijssen, Siegfried; Tack, Guido; De Raedt, Luc
**Main reference** T. Guns, A. Dries, G. Tack, S. Nijssen, L. De Raedt, "MiningZinc: A modeling language for constraint-based mining," in Proc. of the 23rd Int'l Joint Conf. on Artificial Intelligence (IJCAI'13), pp. 1365–1372, AAAI Press, 2013.
**URL** http://www.aaai.org/ocs/index.php/IJCAI/IJCAI13/paper/view/6947
**URL** http://dtai.cs.kuleuven.be/CP4IM/miningzinc/

We presented MiningZinc, a declarative framework for constraint-based data mining. MiningZinc consists of two key components: a language component and an execution mechanism. The language allows for high-level and natural modeling of mining problems, such that MiningZinc models closely resemble the definitions found in the data mining literature. It is inspired by the Zinc family of languages and systems and supports user-defined constraints and functions.

The execution mechanism specifies how to compute solutions for the models. It ensures the solver independence of the language and supports both standard constraint solvers and specialized data mining systems. The high-level problem specification is first translated into a normalized constraint language (FlatZinc). Rewrite rules are then used to add redundant constraints or solve part of the problem specification using specialized algorithms or generic con- straint programming solvers. For one model, different execution strategies are automatically extracted that correspond to different sequences of algorithms to run.

In this way, MiningZinc combines high-level declarative modeling with the performance of state-of-the-art algorithms on well-known tasks. Furthermore, its language allows one to model constraint-based mining problems and varia- tions, and its execution mechanism can detect and re-use existing algorithms. We demonstrate this experimentally on a number of tasks.

### References

1 T. Guns, A. Dries, G. Tack, S. Nijssen, L. De Raedt. MiningZinc: A modeling language for constraint-based mining. In: Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence, AAAI Press, 2013, pp. 1365–1372.

## 3.13 Analysing and Automatically Optimising the Empirical Scaling of Algorithm Performance

*Holger H. Hoos (University of British Columbia – Vancouver, CA)*

**Joint work of** Hoos, Holger H.; Stützle, Thomas; Styles, James

The scaling of running time with input size is of central importance in the analysis and design of algorithms. Traditionally, this is captured by the notion of time complexity and analysed mathematically. However, this analysis is often restricted to simplified or simplistic algorithms, ignores constants and considers only worst- case behaviour, or average-case behaviour under restrictive assumptions on the distribution of inputs. The same limitations apply to algorithm design or selection guided by theoretical time complexity.

In this talk, I first present a new approach for analysing the empirical time complexity of algorithms, that is, the empirical scaling of running time with input size, using a statistically rigorous approach based on resampling statistics. Then, I discuss how automated algorithm configuration can be used to optimise empirical scaling behaviour, using a combination of generic algorithm configuration procedures, racing and statistical testing techniques.

**References**

1 Holger H. Hoos, Thomas Stützle. *On the empirical scaling of run-time for finding optimal solutions to the travelling salesman problem.* European Journal of Operational Research 238:87–94.

2 James Styles, Holger H. Hoos. *Using Racing to Automatically Configure Algorithms for Scaling Performance.* Proceedings of GECCO 2013, pp. 551–558.

## 3.14 Modelling and Optimization of Empirical Algorithm Performance

*Frank Hutter (Universität Freiburg, DE)*

**Joint work of** Hutter, Frank; Hoos, Holger; Leyton-Brown, Kevin; Xu, Lin
**Main reference** F. Hutter, L. Xu, H. H. Hoos, K. Leyton-Brown, "Algorithm runtime prediction: Methods & evaluation," Artificial Intelligence, 206(Jan. 2014):79–111, 2014.
**URL** http://dx.doi.org/10.1016/j.artint.2013.10.003

Algorithm developers and end users often face questions like the following:

- Which parameter setting should I use to optimize my algorithm's empirical performance?
- How does performance depend on the setting of a certain parameter?
- What characteristics distinguish easy from hard problem instances?
- Which of two (or more) available algorithms will perform best on a given new instance?

We describe fully formalized domain-independent methods to answer these questions based on machine learning and optimization techniques. We illustrate the power of these automated methods on a variety of domains, ranging from combinatorial problems (SAT and mixed integer programming) to machine learning (where our methods enable an automatic optimization over the combined space of machine learning algorithms or deep learning architectures and their hyperparameters).

**References**

1 Frank Hutter, Lin Xu, Holger Hoos, Kevin Leyton-Brown (2014). *Algorithm runtime prediction: Methods and evaluation.* Artificial Intelligence Journal (AIJ) 206:79–111.

2 Frank Hutter, Holger Hoos, Kevin Leyton-Brown (2011). *Sequential Model-Based Optimization for General Algorithm Configuration.* Proceedings of the 5th Learning and Intelligent Optimization Conference (LION 2011).

3 Chris Thornton, Frank Hutter, Holger Hoos, Kevin Leyton-Brown (2013). *Auto-WEKA: Combined Selection and Hyperparameter Optimization of Classification Algorithms.* Proceedings of the 19th ACM SIGKDD international conference on knowledge discovery and data mining (KDD 2013).

## 3.15 Relational Linear Programming

*Kristian Kersting (TU Dortmund, DE)*

**Joint work of** Kersting, Kristian; Mladenov, Martin; Tokmakov, Pavel
**Main reference** K. Kersting, M. Mladenov, P. Tokmakov, "Relational Linear Programs," arXiv:1410.3125v1 [cs.AI, 2014.
**URL** http://arxiv.org/abs/1410.3125v1

We propose relational linear programming, a simple framework for combing linear programs (LPs) and logic programs. A relational linear program (RLP) is a declarative LP template defining the objective and the constraints through the logical concepts of objects, relations, and quantified variables. This allows one to express the LP objective and constraints relationally for a varying number of individuals and relations among them without enumerating them. Together with a logical knowledge base, effectively a logical program consisting of logical facts and rules, it induces a ground LP. This ground LP is solved using lifted linear programming. That is, symmetries within the ground LP are employed to reduce its dimensionality, if possible, and the reduced program is solved using any off-the-shelf LP solver. In contrast to mainstream LP template languages like AMPL, which features a mixture of declarative and imperative programming styles, RLP's relational nature allows a more intuitive representation of optimization problems over relational domains. We illustrate this empirically by experiments on approximate inference in Markov logic networks using LP relaxations, on solving Markov decision processes, and on collective inference using LP support vector machines.

## 3.16 The Algorithm Selection Problem: Standard Data Format and Tools

*Lars Kotthoff (University College Cork, IE)*

The Algorithm Selection Problem is attracting increasing attention from researchers and practitioners from a variety of different backgrounds. After decades of fruitful applications in a number of domains, a lot of data has been generated and many approaches tried, but the community lacks a standard format or repository for this data. Furthermore, there are no standard implementation tools. This situation makes it hard to effectively share and compare different approaches and results on different data. It also unnecessarily increases the initial threshold for researchers new to this area.

In this talk, I will present Aslib, a data format specification and benchmark repository for algorithm selection problems. I will then introduce and briefly demonstrate LLAMA, a modular and extensible toolkit implemented as an R package that facilitates the exploration of a range of different portfolio techniques on any problem domain. LLAMA is integrated with Aslib and is able to work with any algorithm selection scenario in the specified data format.

## 3.17 Interactive redescription mining using Siren – a demonstration

*Pauli Miettinen (MPI für Informatik – Saarbrücken, DE)*

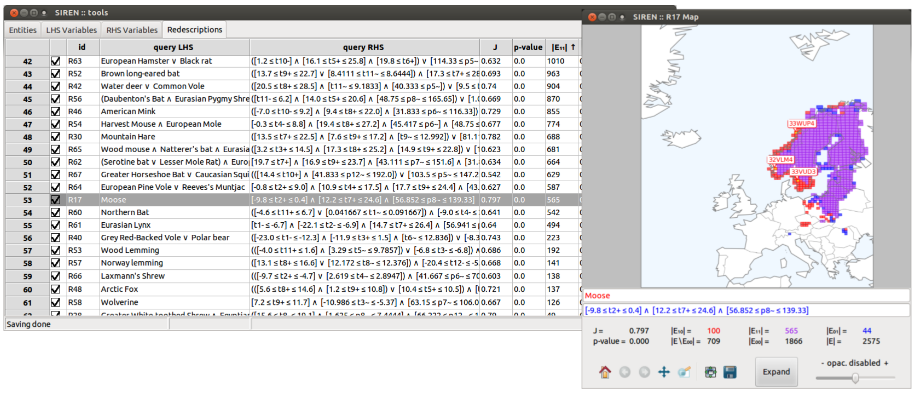
**Joint work of** Galbrun, Esther; Miettinen, Pauli
**Main reference** E. Galbrun, P. Miettinen, "Interactive Redescription Mining," in Proc. of the 2014 ACM SIGMOD Int'l Conf. on Management of Data (SIGMOD'14), pp. 1079–1082, ACM, 2014.
**URL** http://dx.doi.org/10.1145/2588555.2594520

Exploratory data analysis consists of multiple iterated steps: a data mining method is run on the data, the results are interpreted, new insights are formed, and the resulting knowledge is utilized when executing the method in a next round, and so on until satisfactory results are obtained.

We focus on redescription mining, a powerful data analysis method that aims at finding alternative descriptions of the same entities. We demonstrate Siren, a tool for interactive redescription mining. It is designed to facilitate the exploratory analysis of data by providing a seamless environment for mining, visualizing and editing redescriptions in an interactive fashion, supporting the analysis process in all its stages.

Simultaneously, Siren exemplifies the power of the various visualizations and means of interaction integrated into it, techniques that reach beyond the task of redescription mining considered here, to other analysis methods.

Figure 1 shows an example screenshot from Siren. In the left and back, we see a list of redescriptions explaining areas of Europe on one hand by the mammal species that occupy them, and on the other hand, by the areas' bioclimatic conditions (temperature and rainfall). On the right and front, we see one redescription visualised in a map, where purple means areas where both descriptions hold, red means areas where moose lives but the bioclimatic conditions do not hold, and blue means areas where the bioclimatic conditions hold but moose does not live.



| | | id | query LHS | query RHS | J | p-value | $\lvert E_{11}\rvert$ ↑ |
|---|---|---|---|---|---|---|---|
| 42 | ✓ | R63 | European Hamster ∨ Black rat | ([1.2 ≤ t10-] ∧ [16.1 ≤ t5+ ≤ 25.8] ∧ [19.8 ≤ t6+]) ∨ [114.33 ≤ p5~ | 0.632 | 0.0 | 1010 |
| 43 | ✓ | R52 | Brown long-eared bat | ([13.7 ≤ t9+ ≤ 22.7] ∨ [8.4111 ≤ t11~ ≤ 8.6444]) ∧ [17.3 ≤ t7+ ≤ 28 | 0.693 | 0.0 | 963 |
| 44 | ✓ | R42 | Water deer ∨ Common Vole | ([20.5 ≤ t8+ ≤ 28.5] ∧ [t11~ ≤ 9.1833] ∧ [40.333 ≤ p5~]) ∨ [9.5 ≤ t | 0.74 | 0.0 | 904 |
| 45 | ✓ | R56 | (Daubenton's Bat ∧ Eurasian Pygmy Shre | ([t11- ≤ 6.2] ∧ [14.0 ≤ t5+ ≤ 20.6] ∧ [48.75 ≤ p8~ ≤ 165.65]) ∨ [1.0 | 0.669 | 0.0 | 870 |
| 46 | ✓ | R46 | American Mink | ([-7.0 ≤ t10- ≤ 9.2] ∧ [9.4 ≤ t8+ ≤ 22.0] ∧ [31.833 ≤ p6~ ≤ 116.33]) | 0.729 | 0.0 | 855 |
| 47 | ✓ | R54 | Harvest Mouse ∧ European Mole | [-0.3 ≤ t4- ≤ 8.8] ∧ [19.4 ≤ t8+ ≤ 27.2] ∧ [45.417 ≤ p6~] ∧ [48.75 ≤ | 0.677 | 0.0 | 774 |
| 48 | ✓ | R30 | Mountain Hare | ([13.5 ≤ t7+ ≤ 22.5] ∧ [7.6 ≤ t9+ ≤ 17.2] ∧ [t9~ ≤ 12.992]) ∨ [81.1 | 0.782 | 0.0 | 688 |
| 49 | ✓ | R65 | Wood mouse ∧ Natterer's bat ∧ Eurasia | ([3.2 ≤ t3+ ≤ 14.5] ∧ [17.3 ≤ t8+ ≤ 25.2] ∧ [14.9 ≤ t9+ ≤ 22.8]) ∨ [1 | 0.623 | 0.0 | 681 |
| 50 | ✓ | R62 | (Serotine bat ∨ Lesser Mole Rat) ∧ Euro | [19.7 ≤ t7+] ∧ [16.9 ≤ t9+ ≤ 23.7] ∧ [43.111 ≤ p7~ ≤ 151.6] ∧ [31.8 | 0.634 | 0.0 | 664 |
| 51 | ✓ | R67 | Greater Horseshoe Bat ∨ Caucasian Squi | (([14.4 ≤ t10+] ∧ [41.833 ≤ p12~ ≤ 192.0]) ∨ [103.5 ≤ p5~ ≤ 147.2 | 0.542 | 0.0 | 629 |
| 52 | ✓ | R64 | European Pine Vole ∨ Reeves's Muntjac | [-0.8 ≤ t2+ ≤ 9.0] ∧ [10.9 ≤ t4+ ≤ 17.5] ∧ [17.7 ≤ t9+ ≤ 24.4] ∧ [43. | 0.627 | 0.0 | 587 |
| 53 | ✓ | R17 | Moose | [-9.8 ≤ t2+ ≤ 0.4] ∧ [12.2 ≤ t7+ ≤ 24.6] ∧ [56.852 ≤ p8~ ≤ 139.33] | 0.797 | 0.0 | 565 |
| 54 | ✓ | R60 | Northern Bat | ([-4.6 ≤ t11+ ≤ 6.7] ∨ [0.041667 ≤ t1~ ≤ 0.091667]) ∧ [-9.0 ≤ t4- ≤ | 0.641 | 0.0 | 542 |
| 55 | ✓ | R61 | Eurasian Lynx | [t1- ≤ -6.7] ∧ [-22.1 ≤ t2- ≤ -6.9] ∧ [14.7 ≤ t7+ ≤ 26.4] ∧ [56.941 ≤ | 0.64 | 0.0 | 494 |
| 56 | ✓ | R40 | Grey Red-Backed Vole ∨ Polar bear | ([-23.0 ≤ t1- ≤ -12.3] ∧ [-11.9 ≤ t3+ ≤ 1.5] ∧ [t6~ ≤ 12.836]) ∨ [-8.3 | 0.743 | 0.0 | 223 |
| 57 | ✓ | R53 | Wood Lemming | (([-4.0 ≤ t11+ ≤ 1.6] ∧ [3.29 ≤ t5~ ≤ 9.7857]) ∨ [-6.8 ≤ t3- ≤ -6.8]) ∧ | 0.686 | 0.0 | 192 |
| 58 | ✓ | R57 | Norway lemming | ([13.1 ≤ t8+ ≤ 16.6] ∨ [12.172 ≤ t8~ ≤ 12.376]) ∧ [-20.4 ≤ t12- ≤ -5. | 0.668 | 0.0 | 141 |
| 59 | ✓ | R66 | Laxmann's Shrew | (([-9.7 ≤ t2+ ≤ -4.7] ∨ [2.619 ≤ t4~ ≤ 2.8947]) ∧ [41.667 ≤ p6~ ≤ 70 | 0.603 | 0.0 | 138 |
| 60 | ✓ | R48 | Arctic Fox | (([5.6 ≤ t8+ ≤ 14.6] ∧ [1.2 ≤ t9+ ≤ 10.8]) ∨ [10.4 ≤ t5+ ≤ 10.5]) ∧ [t | 0.721 | 0.0 | 137 |
| 61 | ✓ | R58 | Wolverine | [7.2 ≤ t9+ ≤ 11.7] ∧ [-10.986 ≤ t3~ ≤ -5.37] ∧ [63.15 ≤ p7~ ≤ 106.0 | 0.667 | 0.0 | 126 |

Saving done

Moose
[-9.8 ≤ t2+ ≤ 0.4] ∧ [12.2 ≤ t7+ ≤ 24.6] ∧ [56.852 ≤ p8~ ≤ 139.33]

```
J =        0.797   |E10| =      100   |E11| =  565    |E01| =  44
p-value =  0.000   |E \ E00| =  709   |E00| =  1866   |E| =    2575
```

**Figure 1** Screenshot of the Siren program.

## 3.18 Structured Mining Tasks with CP-based frameworks

*Benjamin Negrevergne (KU Leuven, BE)*

**Joint work of** Negrevergne, Benjamin; Dries, Anton; Guns, Tias; Nijssen, Siegfried
**Main reference** B. Negrevergne, A. Dries, T. Guns, S. Nijssen, "Dominance programming for itemset mining," in Proc. of the 2013 IEEE 13th Int'l Conf. on In Data Mining (ICDM'13), pp. 557–566, IEEE, 2013; pre-print available from author's webpage.
**URL** http://dx.doi.org/10.1109/ICDM.2013.92
**URL** http://people.cs.kuleuven.be/~benjamin.negrevergne/documents/2013_icdm_dominance_programming.pdf

A long term goal of the pattern mining community has been to design a unified framework for the variety of pattern mining tasks that are available. Recently De Raedt et al. have proposed to use constraint programming (CP) as a unifying framework and have shown that it can be used to elegantly formalize and solve many itemset mining tasks (pattern mining, where patterns are sets). Whether the same technique can be used to formalize more complex mining tasks such as sequence or graph mining is a topic of interest.

In this talk we present our experience with constraint programming for sequence mining. We show that although basic sequence mining tasks can be formalized in constraint programming, the resulting models suffer from several limitations that are not easy to solve in the CP framework. To address these problems, we introduce a more general framework called relational constraint programming which combines ideas from constraint programming and ideas from the relational algebra. Relational constraint programming is a language that resemble the relational algebra but uses constraint programming techniques for evaluating the queries. Finally we show many sequence mining problems can be elegantly formalized and solved in relational constraint programming.

## 3.19 Relational Constraint Programming

*Siegfried Nijssen (Universiteit leiden, NL; KU Leuven, BE)*

**Joint work of** Negrevergne, Benjamin; Dries, Anton; Guns, Tias; Nijssen, Siegfried
**Main reference** B. Negrevergne, A. Dries, T. Guns, S. Nijssen, "Dominance programming for itemset mining," in Proc. of the 2013 IEEE 13th Int'l Conf. on In Data Mining (ICDM'13), pp. 557–566, IEEE, 2013; pre-print available from author's webpage.
**URL** http://dx.doi.org/10.1109/ICDM.2013.92
**URL** http://people.cs.kuleuven.be/~benjamin.negrevergne/documents/2013_icdm_dominance_programming.pdf

This talk argues that a significant number of data mining and machine learning problems can not be formalized as pure constraint satisfaction or optimization problems. Many problems also require a definition of *preference* between solutions, require some form of *aggregation*, and require that some variables ignored. To formalize and solve such problems well-known constraint programming systems, such as Numberjack and the MiniZinc toolchain, do not provide the required support. We propose a new formalism, relational constraint programming, together with solution strategies for this formalism, that do allow for formalizing and solving this wide range of data mining and machine learning problems.

## 3.20 Structured Learning Modulo Theories

*Andrea Passerini (University of Trento – DISI, IT)*

**Joint work of** Passerini, Andrea; Paolo, Campigotto; Roberto, Battiti; Stefano, Teso; Roberto, Sebastiani
**Main reference** S. Teso, R. Sebastiani, A. Passerini, "Structured Learning Modulo Theories," arXiv:1405.1675v2 [cs.AI], 2014.
**URL** http://arxiv.org/abs/1405.1675v2

Modelling problems containing a mixture of Boolean and numerical variables is a long-standing interest of Artificial Intelligence. However, performing inference and learning in hybrid domains is a particularly daunting task. The ability to model this kind of domains is crucial in 'learning to design' tasks, that is, learning applications where the goal is to learn from examples how to perform automatic de novo design of novel objects. In this talk I will present Structured Learning Modulo Theories, a max-margin approach for learning in hybrid domains based on Satisfiability Modulo Theories, which allows to combine Boolean reasoning and optimization over continuous linear arithmetical constraints. I will present a number of artificial and real world scenarios showing the potential of the approach.

## 3.21 Probabilistic Conditional Preferences

*Francesca Rossi (University of Padova, IT)*

**Joint work of** Cornelio, Cristina; Goldsmith, Judy; Grandi, Umberto; Mattei, Nicholas; Rossi, Francesca; Venable, Brent

PCP-nets generalise CP-nets to model conditional preferences with probabilistic uncertainty. In this paper we use PCP-net to compactly model a collection of CP-nets, thus using probabilistic uncertainty to reconcile (possible conflicting) preferences expressed by a group of agents. We then study two main tasks: finding an optimal outcome which best represents the preferences of the group of agents, and answering dominance queries. Theoretical and experimental analysis allows us to find efficient and accurate algorithms to perform both tasks.

## 3.22 Building bridges between data mining and constraint programming

*Lakhdar Sais (Artois University – Lens, FR)*

**Joint work of** Jabbour, Said; Sais, Lakhdar; Salhi, Yakoub; Takeaki, Uno
**URL** http://www.cril.univ-artois.fr/decMining/

This talk, we overview our contribution to data mining and more generally to the cross-fertilization between data mining, constraint programming and propositional satisfiability (http://www.cril.univ-artois.fr/decMining/). We will focus on two contributions. First, we show how propositional satisfiability can be used to model and solve problems in data mining. As an illustration, we present a SAT-based declarative approach for enumerating top-k (closed, frequent) itemsets in transactional databases. Secondly, we discuss the potential contribution of data mining to propositional satisfiability. In this context, we present a first application of data mining to compress Boolean formulas conjunctive normal form.

This work is supported by the ANR DAG project "Declarative approaches for enumerating interesting patterns" (http://liris.cnrs.fr/dag/).

**References**

1 Said Jabbour and Lakhdar Sais and Yakoub Salhi. *The Top-k Frequent Closed Itemset Mining Using Top-k SAT Problem.* In proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases ECML/PKDD'13, pp. 403–418, Prague, LNAI 8188, Springer, September 23–27, 2013.

2 Said Jabbour and Lakhdar Sais and Yakoub Salhi and Takeaki Uno. *Mining-based Compression Approach of Propositional Formulae.* In Twenty-Second ACM International Conference on Information an Knowledge Management CIKM'2013 , pp. 289–298, October 27th to November 1st, San Francisco, 2013.

## 3.23 C10: Timed, Probabilistic (concurrent) constraint programming

*Vijay A. Saraswat (IBM TJ Watson Research Center – Hawthorne, US)*

Over the last twenty years, the theory of concurrent constraint programming has seen significant progress. CCP offers a declarative framework (in the "computation as deduction" paradigm) for concurrent, constraint based computation that subsumes backward-chaining (pure Prolog) and forward- chaining computations, timed computations, and probabilistic computations. We are now developing C10, a concrete language in this framework intended to be compiled into (concurrent, distributed) X10, and support combinatorial problem solving and (probabilistic) analytic computations on big data.

## 3.24 Learning Constraint Value

*Michele Sebag (University of Paris South XI, FR)*

Active Learning (AL) and Experiment Design (ED) are problems where one is looking for constraints: these constraints serve to prune the space of candidate hypotheses.

The constraint is made of a question, and the oracle's answer: the oracle gives the label of the instance (AL), or runs the experiment and reports the behavior of the system under identification (ED).

When the oracle is expensive, you are given a budget $k$: find the most informative $k$ instances /experiments.

This talk discussed:

- how to formalize active learning and experiment design on a budget as reinforcement learning problems (learn to sequentially sample the instance or the experiment space)
- how to find an approximation of the optimal sampling strategy, using Monte-Carlo Tree Search.

## 3.25 Constraints To Specify Data

*Arno Siebes (Utrecht University, NL)*

Patterns one observes in a data set can be seen as constraints on possible further data sets one samples from the same distribution. For example, if the vast majority of your customers buy both diapers and beer today, you expect them to do the same tomorrow. Different from normal constraint problems, however, not all solutions – i. e., data sets that satisfy a given set of constraints – are equally good. For example, if our constraints tell us that the data should be smaller than 1 and bigger than $-1$, a data set that only has values in $[0, 1]$ is not a very satisfying solution; it is not typical for the set of all solutions.

The obvious solution to this problem is to define a probability distribution on the set of all possible data sets such that the more typical data sets have a high probability. Since the collection of all possible data sets is rather large, we do not define the probability explicitly, but implicitly through a generative model, i. e., a model that can generate each possible data set.

The models we consider consist of a small number of of pairs $(m_i, p_i)$ in which $m_i$ is a pattern and $p_i$ is (proportional to) the probability that $m_i$ is used in the generation of a tuple in a new data set. Stipulating that the elements of a model are independent than leads to the simple generative procedure of picking elements proportional to the $p_i$.

The quality of such a generative model $M$ lies in how likely it is to generate a data set $D'$ that resembles the original data set $D$. That is, if we use $M$ to generate, say, 1000 data sets and we "hide" $D$ in that collection: how easy is it to identify $D$. If we can find $D$ is (considerably) less than a logarithmic number of steps, $M$ is a bad model otherwise it is a good model. This informal idea can be formalised through the Minimum Description Length principle.

As usual in MDL, our goal is to find the smallest good model, but algorithms to achieve this goal are not part of this talk.

## 3.26 OpenML: Networked science in machine learning

*Joaquin Vanschoren (TU Eindhoven, NL)*

**Joint work of** Vanschoren, Joaquin; van Rijn, Jan N.; Bischl, Bernd; Torgo, Luis
**Main reference** J. Vanschoren, J. N. van Rijn, B. Bischl, L. Torgo, "OpenML: networked science in machine learning," SIGKDD Explorations, 15(2):49–60, 2013.
**URL** http://www.kdd.org/sites/default/files/issues/15-2-2013-12.pdf
**URL** http://www.openml.org

Today, the ubiquity of the internet is allowing new, more scalable forms of scientific collaboration. Networked science uses online tools to share and organize data on a global scale so that scientists are able to build directly on each other's data and techniques, reuse them in unforeseen ways, and mine all data to search for patterns.

OpenML.org is a place to share and reuse machine learning data sets, tools and experiments. It offers web services for easy integration in many machine learning systems and is readily integrated in R, Weka, Rapidminer and others. It helps researchers win time by reusing and automating machine learning experiments, and gain more credit for their work by making it more easily discoverable and easily reusable.

Moreover, OpenML helps scientists and students to explore different machine learning techniques, find out which are most useful in their work, and collaborate with others to analyze scientific data online. OpenML can itself also learn from the experiments of many researchers, and use algorithm selection and optimization techniques to help people make better use of machine learning in their work.

### 3.27 Correlation Constraints

*Toby Walsh (NICTA – Sydney, AU)*

Correlation constraints ensure that solutions are correlated or uncorrelated. I motivate the introduction of correlation constraints, and discuss how they can be propagated. Applications including security games where we wish schedules to be uncorrelated, and logistics where we may wish tomorrow's route to be correlated with today's. Whilst computing the correlation between two solutions is easy, propagation of correlation constraints can be more intractable to compute as we are essentially computing all solutions.

## 4 Working Groups

### 4.1 Declarative Languages for Machine Learning and Data Mining

*Ian Davidson (UC Davis, USA), Luc De Raedt (KU Leuven, BE), Siegfried Nijssen (KU Leuven, BE; and Universiteit Leiden, NL)*

Within the seminar, there was a strong interest in declarative languages for machine learning and data mining. Existing machine learning and data mining algorithms are almost always implemented in procedural languages and a natural evolution is for their implementation in a declarative language. A discussion group was formed that was focused on the following dimensions:

- *General vs. specific languages.* It was agreed upon that generic languages were the focus of the discussion group; within such languages, the focus was on identifying the primitives necessary for allowing mining and learning.
- *Modeling vs. programming.* The focus of the discussion group was on declarative modeling languages that allow to specify a mining or learning task, such as minimum square error clustering; programming languages, which allow to specify execution steps, such as those of the $k-$means algorithm, were not considered.
- *Languages vs. environment.* The focus of the discussion group was on languages. Although it was agreed that a programming environment that provides feedback to the programmer, such as a debugger, is useful, it was felt that the scope would be too broad when including environments in the discussion.

The group first identified why a declarative language for data mining and machine learning is desirable. Several benefits were identified for declarative modeling languages:

- these languages allow for quick modeling of problems;
- these languages allow for quick prototyping of solution methods;
- these languages require less lines of code to specify and solve a task;
- these languages allow for solving tasks for which currently no specialized algorithm exists;
- these languages encourage the reproducibility of results;
- these languages force one to think about and identify general principles;
- programmers can learn from the formal problem specifications and use these specifications when developing specialized solvers;
- the primitives within such languages could categorize the area; one could think of building a catalogue of constraints;
- the language can be used as a specification language: one could check later whether results of specialized algorithms are correct for certain specifications.

A key aspect of the modeling language is the choice of primitives that it supports. Possible primitives were considered to be those for specifying:

- Optimization criteria (loss functions) that capture the underlying learning problem;
- The calculation of Kernels and distances;
- Statistical assumptions, for instance, in probability distributions;
- Constraints and preferences that can encode human guidance and problem constraints;
- The underlying tasks (structure of the problem), including the hypothesis (or function) space and the structure of the output.

A lively discussion was held involving the last aspect.

When restricting the attention to machine learning, two perspectives were considered. One is the one in which machine learning is formalized as finding a function in a hypothesis space such that a loss function is optimized. Each function in the hypothesis space is a function that maps an instance in an input space to an element in an output space, such as for example a class label. A language for machine learning in this case should include primitives for specifying a hypothesis space and a loss function.

The other perspective is that in which machine learning is formalized as a constraint optimization problem, that is, finding an assignment to variables such that constraints and optimization critera are satisfied on these variables. Primitives in such a language should allow to specify variables, their domains, constraints and optimization criteria.

Even though differently focused, it was concluded that these perspectives may be mapped to each other: the variables and domains in one perspective correspond to a hypothesis space in the other perspective, while the loss function corresponds to the optimization criterion. Input and output spaces are encoded across variables, domains, constraints and optimization criteria.

Another discussion revolved around the differences between machine learning and data mining. The argument was made that machine learning and data mining differ from each other as machine learning is specifically focused on finding functions from input to output spaces, while data mining does not always have an output space. This lack of output space reflects the more exploratory nature of data mining.

An inventarisation was made of tasks that could be studied using a declarative data mining and machine learning language. Simple tasks that have already been studied include $k-$means clustering, spectral clustering, itemset mining and decision tree induction.

More difficult tasks include: manifold learning, learning SVMs, structured output prediction, graph clustering or frequent graph mining and label propagation.

As a challenge it was identified to model a number of these problems in existing declarative modeling languages. The following representative problems were picked:

- frequent graph mining;
- learning to rank;
- structured output prediction;
- label propagation.

The data mining and machine learning members of the discussion group agreed to provide written specifications of the problems, while the constraint programming members agreed to formalize these problems in existing modeling languages. This challenge will continue after the Dagstuhl seminar. It is expected that these tasks will pose various problems for existing modeling systems. The discussion group is expected to continue over e-mail after the seminar.

## 4.2 Learning and Optimization with the Human in the Loop

*Andrea Passerini (University of Trento – DISI, IT), Michele Sebag (Université Paris Sud, FR)*

### 4.2.1 Preamble

This discussion group actually merged two topics: Learning and Optimization (L & O, Andrea Passerini) and Human in the Loop (H-I-L, Michele Sebag).

### 4.2.2 Formal background

Let $\mathcal{X}$ define a solution space. Let $f$, the objective function to be optimized, be defined on $\mathcal{X}$. $f$, referred to as scoring function or quality function, is usually defined from $\mathcal{X}$ onto $\mathbb{R}$; occasionally it can be defined onto $\mathbb{R}^d$ (multi-objective optimization: quality, robustness, cost).

In L & O, $f$ is mostly learned from data (with possible feedback from the human in the loop); in H-I-L, $f$ is mostly learned from the the human in the loop.

In all applications, the goal is twofold: learn $f$; find $\arg\max f$. Depending on the applications, the stress is put on the learning or on the optimization task.

Function $f$ is assumed to abstract all there is to know about the decision problem problem details: decision variables, domains, constraints, objectives preferences.

#### 4.2.2.1 Position of the problem

ILO tackles a sequential decision problem: in each iteration

- The ILO agent must build a hypothesis from the available evidence;
- Based on this evidence, it must act (reinforcement learning) and/or ask for additional evidence (active learning) and/or achieve model-based optimization (active optimization).

#### 4.2.2.2 Theory

- Is there a functional gradient that the ILO trajectory follows (as in boosting)?
- Currently, the criteria leading the ILO trajectory are based on: Expected Global Improvement and Continuous Bayesian Optimization [17, 4, 12, 6]; Expected Posterior Utility [16].

### 4.2.3 Possible applications

- Automated algorithm selection and configuration [9, 8]
- Structured recommendation, optimal design, computational creativity [15]
- Agents/robots (policy learning)

### 4.2.4 Categories of problems

#### 4.2.4.1 Nature of the ILO input

Feedback to the agent can be provided by a user, some experimental procedure, or the environment itself. Depending on the source, different types of feedback can be conceived:

- quality of candidate solution
- preferences between candidate solutions [16, 11, 5, 1, 18, 10]
- explanations on why a certain solution is not satisfactory
- sketches of desired solution
- feedback on partial solutions, or repairs a solution proposed by the system [11, 10]
- comments on parts of the model [2]
- partial reward

#### 4.2.4.2 Representation of the solution space

Different degrees of complexity can be identified for the solution space, depending on the type of application which can be conceived:

**homogeneous vector** in the simplest case the solution space is $\mathcal{X} = \mathbb{R}^D$: the goal is to find some optimal vector $x$ in $\mathbb{R}^D$. This is the typical setting in interactive optimization.

**heterogeneous vector** a more complex setting is when the solution space is restricted according to variable dependencies (e. g. some dependent variables are relevant only for certain values of the variables they depend on). This can be thought of a solution space $\mathcal{X} = \mathbb{R}^D \times \{0,1\}^{D'}$ where variable dependencies impose restrictions on the relevant portions of the space. An application example is algorithm configuration [9, 7].

**structure** many optimization tasks require to return a structured object, with possibly both continuous and discrete variables, and relations between them. Application scenarios are in recommendation, optimal design (e. g. in chemistry, molecule maximizing a given property), marketing (bundle of related items – sketch of a house). Here the space of functions $f$ can be formalized in terms of stochastic graph grammars (less expensive, a stochastic context free grammar) or constrained optimization problems.

**function** in reinforcement learning, the solution space is a functional space. The sought solution is a policy, i. e. a function mapping any state onto an action. Note that the state abstracts everything relevant to decision.

#### 4.2.4.3 Representation of the objective function space

The type of representation for the objective function depends on the type of expected interaction with the user:

- In the white box case, the user can inspect $f$ and provide explicit comments/repair/debug.
- In the black box case, the user only provides feedback on the solution proposed by the ILO agent.
- In the gray box case, the user can comment on parts of the solution, or on parts of $f$.

#### 4.2.4.4 Learning the representations

If a dataset including real solutions is available, it is possible to learn constraints (e. g. learning from positive only). Another possibility, inspired from continuous language model [3], is to use supervised learning and discriminate actual solutions from lesioned ones. Unsupervised learning techniques can be also employed to learn useful internal representations for the solutions (representation learning). These approaches aim at learning explicit or implicit constraints on $f$. However, learning $f$ requires some supervised input (feedback).

### 4.2.5 At the crossroad of ML and Computer Human Interaction

Some issues are at the cross-road of ML and CHI.

#### 4.2.5.1 User profiling

The resolution of the ILO problem will firstly depend on who is the human in the loop and what is her profile: digitally proficient or naive, wants a solution or wants an explanation or both; wants no burden, etc.

#### 4.2.5.2 Human bias

The user's feedback is bound to be noisy (even more so if preference drift is taking place). There is a systematic bias (manifest e. g. in Robotics application; e. g. Thomaz et al. mention that some users say "it's good" even when undeserved, to "encourage" the agent :-) )

#### 4.2.5.3 Non stationarity

Related to the two above is the fact that the user might change her mind along the process (preference drift).

#### 4.2.5.4 The Crowd in the Loop: CIL

There might be more than one user. Having a crowd in the loop increases the noise.

#### 4.2.5.5 ILO on a budget

Querying the user is analogous to optimizing an expensive function [14]. You might want to limit a priori the number of queries, the number of experiments [13], etc. Related questions:
- What is the stopping criterion (when the user says so?)
- How far are we from the goal?

### 4.2.6 Interface and friendliness

The ILO agent should not jump here and there in the solution space: the queries must display some stability; the user must be and feel in control. Still, experiments in robotics suggest that the process is more efficient if the learning partner is leading the interaction.

### 4.2.7 Autonomy and Safety

Two settings must be distinguished.
1. The safe case is when the ILO agent can act and ask the user's (or the world's) feedback.
2. The critical case is when the ILO agent must ask for feedback first and act if allowed.

Many cases are in the intermediate region: the ILO agent should ask for feedback in the early interaction phase; but at some points it should know when it can act autonomously.

## References

1 Riad Akrour, Marc Schoenauer, Michèle Sebag, and Jean-Christophe Souplet. Programming by Feedback. In *International Conference on Machine Learning*, Pékin, China, June 2014.

2 Ch. Bessiere, R. Coletta, E. Hebrard, G. Katsirelos, N. Lazaar, N. Narodytska, C.-G. Quimper, and T. Walsh. Constraint acquisition via partial queries. In . Rossi, editor, *Proc. IJCAI.* IJCAI/AAAI, 2013.

3 Antoine Bordes, Léon Bottou, Ronan Collobert, Dan Roth, Jason Weston, and Luke Zettlemoyer. Introduction to the special issue on learning semantics. *Machine Learning*, 94(2):127–131, 2014.

4 Eric Brochu, Nando de Freitas, and Abhijeet Ghosh. Active preference learning with discrete choice data. In *Advances in Neural Information Processing Systems*, 2007.

5 J. Fürnkranz, E. Hüllermeier, W. Cheng, and S.-H. Park. Preference-based reinforcement learning: a formal framework and a policy iteration algorithm. *Machine Learning*, 89(1–2):123–156, 2012.

6 Roman Garnett, Yamuna Krishnamurthy, Xuehan Xiong, Richard Mann, and Jeff G. Schneider. Bayesian optimal active search and surveying. In John Langford and Joelle Pineau, editors, *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, pp. 1239–1246, New York, NY, USA, 2012. ACM.

7 Holger H. Hoos. Programming by optimization. *Commun. ACM*, 55(2):70–80, February 2012.

8 F. Hutter, H. Hoos, and K. Leyton-Brown. An efficient approach for assessing hyperparameter importance. In *Proc. of ICML-14*, 2014. To appear.

9 Frank Hutter, Lin Xu, Holger H. Hoos, and Kevin Leyton-Brown. Algorithm runtime prediction: Methods & evaluation. *Artificial Intelligence*, 206(0):79–111, 2014.

10 A. Jain, B. Wojcik, T. Joachims, and A. Saxena. Learning trajectory preferences for manipulators via iterative improvement. In *Neural Information Processing Systems (NIPS)*, pp. 575–583, 2013.

11 W. Bradley Knox, Peter Stone, and Cynthia Breazeal. Training a robot via human feedback: A case study. In *Social Robotics*, October 2013.

12 Daniel James Lizotte. *Practical Bayesian Optimization.* PhD thesis, Edmonton, Alta., Canada, 2008. AAINR46365.

13 A. Llamosi, A. Mezine, F. D'Alché-Buc, V. Letort, and M. Sebag. Experimental design in dynamical system identification: A bandit-based active learning approach. In *ECML PKDD*, volume 8725-2, pp. 306–321, 2014.

14 I. Loshchilov, M. Schoenauer, M. Sebag, and N. Hansen. Maximum Likelihood Online Adaptation of Hyper-parameters in CMA-ES. In Th. Bartz-Beielstein et al., editor, *PPSN XIII*, pp. 70–79. LNCS 8672, Springer Verlag, 2014. Best Paper Award.

15 Andrea Passerini Thomas Gärtner, Roman Garnett, editor. *Constructive Machine Learning*, 2013.

16 Paolo Viappiani and Craig Boutilier. Optimal bayesian recommendation sets and myopically optimal choice query sets. In *Advances in Neural Information Processing Systems 23 (NIPS).* MIT press, 2010.

17 J. Villemonteix, E. Vázquez, M. Sidorkiewicz, and E. Walter. Global optimization of expensive-to-evaluate functions: an empirical comparison of two sampling criteria. *J. Global Optimization*, 43(2-3):373–389, 2009.

**18** Aaron Wilson, Alan Fern, and Prasad Tadepalli. A bayesian approach for policy learning from trajectory preference queries. In F. Pereira, C.J.C. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pp. 1133–1141. Curran Associates, Inc., 2012.

## 4.3 Meta-Algorithmic Techniques

*Holger Hoos (University of British Columbia, CA)*

Meta-algorithmic techniques are methods that operate on algorithms with the goal to achieve performance improvements on specific classes of inputs (or problem instances). Examples for such techniques are algorithm configuration, algorithm selection, algorithm scheduling, parallel algorithm portfolios and restart strategies. For all of these, there are per-set (or per-distribution) and per-instance variants, as well as offline, online and dynamic/adaptive variants. Closely related to such meta-algorithmic design techniques are performance prediction methods, which enable many of them, particularly selection.

Apart from clarifying concepts and terminology, the session focussed on compiling an overview of existing techniques, some discussion of what users actually want, challenges and open questions, and directions for future work. In the following, I give a brief overview of each of these topics.

### Existing techniques and systems

#### General per-set algorithm configuration systems

- ParamILS (based on stochastic local search), see http://www.cs.ubc.ca/labs/beta/Projects/ParamILS/
- I/F-Race (based on iterative racing), see http://iridia.ulb.ac.be/irace/
- GGA (based on a genetic algorithm), see https://wiwi.uni-paderborn.de/dep3/entscheidungsunterstuetzungssysteme-und-operations-research-jun-prof-dr-tierney/research/source-code/#c37765
- SMAC (based on sequential model-based optimisation), see http://www.cs.ubc.ca/labs/beta/Projects/SMAC/

Of these, SMAC is the only general system that is based on machine learning, in the sense that it builds a model that can predict performance on previously unseen parameter settings. There are similar approaches that are more limited to a machine learning context, e. g. https://github.com/berndbischl/mlrMBO, aspects of which might perhaps be integrated into future general configuration procedures. AClib (http://www.aclib.net is a useful library of algorithm configuration scenarios, which provides a single interface to ParamILS, I/F-Race and SMAC.

#### Per-instance configuration systems

- ISAC (IBM) – not publically available due to licensing issues; earlier version available as Matlab and R implementation, see https://sites.google.com/site/yurimalitsky/downloads.
- Hydra (UBC) – not currently publicly available (but public release is planned for the near future)

**Algorithm selection systems**

- SATzilla (UBC), see http://www.cs.ubc.ca/labs/beta/Projects/SATzilla/; currently only publicly available in a Matlab implementation that is hard to use; a more usable Java implementation should be available soon.
- CSHC (IBM) – not publically available due to licensing issues;
- LLAMA, see https://bitbucket.org/lkotthoff/llama
- Claspfolio, see http://www.cs.uni-potsdam.de/claspfolio/
- SNNAP, see https://sites.google.com/site/yurimalitsky/downloads
- ARS, see http://hal.archives-ouvertes.fr/docs/00/92/28/40/PDF/techrep_ARS.pdf; system should be available soon, could be extended to instance-specific configuration.

ASlib (http://www.aslib.net) is a library of algorithm selection scenario and simple baseline algorithms.

**Parallel portfolio systems**

- parHydra – currently not publically available (but this may change in the future)

**Online learning systems**

- Continuous Search in Constraint Programming (Arbelaez, Hamadi, Sebag, ICTAI 2010); also Alejandro Arbelaez' PhD thesis
- Jan N. van Rijn, Geoffrey Holmes, Bernhard Pfahringer, Joaquin Vanschoren:Algorithm Selection on Data Streams. Discovery Science 2014:325–336.

**Lifelong learning systems**

- REACT – currently not publically available
- elSAC – currently not publically available

*Note:* Offline algorithm configuration can be used for online configuration (lifelong learning) by collecting instances over time and using an offline configurator on the resulting sets, perhaps using some expiry or down-weighting mechanism for older instances.

**Parameter importance** techniques that can be used to assess which of a given set of parameters of an algorithm impact performance most, individually or in combination with others

- Functional ANOVA, see http://www.automl.org/fanova – designed to be easy to use.
- Ablation analysis, see http://www.cs.ubc.ca/labs/beta/Projects/Ablation/
- Forward selection, see http://www.cs.ubc.ca/labs/beta/Projects/EPMs/ – currently only Matlab code, quite difficult to use.

## What users want

Two example scenarios were discussed that illustrate some aspects of what users of meta-algorithmic techniques might be particularly interested in:

- Given a SAT instance, run the best SAT parameterized algorithm for this. Currently only available for algorithm selection (e. g., SATzilla as it ran in the SAT competition).
- Lifelong learning: improve performance over time (see links to systems above).

## Research questions and challenges

The following research questions and challenges were identified:

- How to produce insight into given algorithms and problems, using meta-algorithmic techniques?
- Better features – there is motivation for domain experts to develop these features, e. g., impact on performance, insights into what works where and why.
- Probing features for big data: based on subsets of data, short runs on data, . . .
- Given a problem $X$, encoded into $Y$ and solved as $Y$, should we use problem features from $X$, from $Y$, or both? (See: Proteus: A Hierarchical Portfolio of Solvers and Transformations, Hurley, Kotthoff, Malitsky, O'Sullivan, CPAIOR 2014)
- Training data is very important; we need better principles for building / selecting effective training sets for configuration, selection, etc.
- How to measure how good a training set is (as a training set)? Once we can measure this, we might be able to optimise it.

### Automated modelling

Many formalisms (MIP, CSP, SAT, SMT, ASP, . . . ) permit various representations or encodings of a given problem. Automating this modelling tasks is an important topic, and the group spent some time discussing existing work and open questions in this areas.

- Dominion (St. Andrews) is a generic system that supports automated modelling; it is quite powerful, but at this point quite difficult to use. It currently does not use much machine learning.
- When is reformulation or transformation needed? How should it best be done? (Additional constraints? Transfer to different modelling paradigm, e. g., MIP to CSP?)
- To which extent can existing configuration / selection techniques be used to help with modelling?
- To which extent is modelling akin to search in a combinatorial space?
- Important: symmetry breaking at the model and instance level
- For MIP: a central goal in reformulations is to obtain tighter linear relaxations (should draw on existing work on reformulations, for example Sherali-Adams reformulations, other work on extended formulations; also Dantzig-Wolfe, see http://www.or.rwth-aachen.de/gcg/
- Spiral: synthesis for Fourier transforms (its space contains the approaches from hundreds of journal papers), see http://www.spiral.net/software/sfft.html
- Franz Brglez (http://www.csc.ncsu.edu/people/brglez) demonstrated that in SAT, renaming variables / reordering clauses can cause major performance loss in high-performance DPLL algorithms.
- Alternatives / variants of models should be evaluated on multiple solvers / solvers configured for those (to avoid bias by solver / solver configuration)
- Potentially very interesting: model selection.

Overall, participants in the session felt that meta-algorithmic techniques were already very useful, but poised to have even larger impact; there was also a general sense that much interesting work remained to be done in this area.

## 4.4 Big Data

*Siegfried Nijssen (KU Leuven, BE; and Universiteit Leiden, NL), Barry O'Sullivan (University College Cork, IE)*

This working group considered the relationships between big data, data mining and constraint programming.

### 4.4.1 What is Big Data?

The first question studied was "what is big data?". IBM have proposed four useful dimensions on which to characterise big data: variety (heterogeneity), volume (scale of data), veracity (uncertainty of data), and velocity (streaming data).[3] Therefore, big data presents a major challenge to data scientists. Methods must capable of working at scale both from a volume and speed perspective and capable of reasoning under uncertainty. Heterogeneity presents significant integration issues. Big data also presents major human-computer interaction challenges to facilitate understanding; explanation and visualisation technologies are, therefore, a major opportunity.

### 4.4.2 Applications

The list of big data applications discussed included:

- Systems biology: e. g., understanding the differences between organisms.
- Pedigree Learning: understanding relationships between organisms.
- Hyper-spectral imaging: huge amounts of event data.
- Radio-astronomy applications: for instance, 800ns is the minimum frequency for studying gravitational waves, and produces massive amounts of data.
- Twitter: geo-positional data, data about spread of viruses, or learning about preferences or reputation.
- Crowd-sourcing: campaigns where non-experts are asked to enrich data from experts, e. g. biodiversity campaigns.
- Peace-keeping and humanitarian applications: data that is collected as a result of the Nuclear Test-ban Treaty, to measure isotopes in the air, ultra-sound, seismographs, etc.; what makes this data interesting is also that different sensors are measuring at different speeds.
- Carpooling, GPS, etc.
- Machine translation.
- Physics: data such as from the ice-cube project, measuring neutrinos, and data from CERN, most of which will be analyzed after the accelerators have stopped.
- Data from the sky-survey project.
- Data collected for predictive maintenance in factories, bridges, etc.

[3] http://www.ibmbigdatahub.com/tag/587

### 4.4.3 Privacy and Ethical Issues

There was considerable discussion around privacy and ethical issues. It was observed that anonymising data sets is very difficult due to the availability of data that can be joined and cross-referenced. The example was given of a join of the Netflix and IMDB databases. A particularly interesting example related to the "How Unique are You?" project at Harvard.[4] That project showed how almost 90% of the population of one particular US state could be uniquely identified by combining two publicly available data-sets.

An important issue is that of adversary models: what do you want to keep private, and what is the adversary who wishes to violate one's privacy assumed to be capable off?

### 4.4.4 Techniques

A discussion focused on techniques that can be applied on big data. One question that was raised was which complexity bounds should hold on big data. Promising methods that were mentioned are based on *core-point analysis*. Within a dataset these methods determine core points, and only these core points are fed into a standard algorithm.

An alternative is to use a sampling approach.

The main application in which both techniques do not work is that of outlier detection due to computational complexity issues.

### 4.4.5 Data & Optimization

Several topics were discussed at the intersection of big data and optimization:

- Finding a good visualization is an optimization problem.
- Finding a good summarization of big data is an optimization problem.
- Increasingly, also optimization models can be big, leading to 'big models', which pose a challenge for current solvers.
- The creation of CSPs may involve combining a large number of different data sources (e. g., from the Internet), and hence the use of big data. See for instance Open Constraint Satisfaction by Faltings et al. In such big data contexts, solution techniques become more 'repair'-based, e. g. local search, large neighbourhood search; these techniques can also operate in parallel.

## 4.5 Towards Killer Applications

*Siegfried Nijssen (KU Leuven, BE; and Universiteit Leiden, NL), Barry O'Sullivan (University College Cork, IE)*

### 4.5.1 What is a Killer Application?

The first discussion item was: what is a killer application? Two definitions may be used:

- Something that is used by a large population of people, but is not necessarily a sophisticated tool addressing a complex problem.
- An application that has a large impact on a well-defined area.

[4] http://aboutmyinfo.org/index.html

### 4.5.2 Examples of Killer Applications

These were discussed as examples of well-established killer applications of current machine learning, data mining and optimization technology:

- Predicting disease transmission from geo-tagged micro-blog data, as performed by Henry Kautz, Adam Sadilek, Henry A. Kautz, Vincent Silenzio (AAAI, 2012).
- Sudoku, since it has had a large impact on creating an awareness of reasoning about constraints amongst the general population.
- Optical character recognition.
- Spam filtering.
- Google machine translation.
- Recommender systems on websites.
- Massive open online courses.
- Automatic tagging of photos (for machine learning), and associated with it: retrieve by content (what it was) rather than address (where you put it) and personal information management.
- User modelling.

### 4.5.3 Competitions

Competitions can be useful to encourage progress in an area. Examples of such competitions that were deemed to have had major impact included:

- competitions run by ChaLearn, http://www.chalearn.org/.
- competitions on Kaggle, http://www.kaggle.com.
- the configurable SAT Solver Challenge, http://aclib.net/cssc2014/.
- the AutoML Challenge (run by ChaLearn), http://codalabtest.cloudapp.net/competitions/762.

### 4.5.4 What Applications would have a Massive Effect (in 5 years)?

Applications with such an impact might be:

- Applications for protecting/managing personal privacy.
- Evidence-based medicine.
- Platforms for facilitating the expression of preferences, preference aggregation, learning over time, collaboration.
- Robot scientists.
- Sports, including video game tournaments.
- Synthetic literature and music generation.
- Advertising.
- Smart homes.
- Alhzeimers treatment planning.
- Companions for the elderly – robotic companions.
- Complex services – air traffic.
- Autonomous cars and traffic management.
- Computational Sustainability.

## Participants

- Hendrik Blockeel
KU Leuven, BE
- Jean-François Boulicaut
INSA – Lyon, FR
- Ken Brown
University College Cork, IE
- Bruno Crémilleux
Caen University, FR
- James Cussens
University of York, GB
- Krzysztof Czarnecki
University of Waterloo, CA
- Thi-Bich-Hanh Dao
University of Orleans, FR
- Ian Davidson
Univ. of California – Davis, US
- Luc De Raedt
KU Leuven, BE
- Marc Denecker
KU Leuven, BE
- Yves Deville
University of Louvain, BE
- Alan Frisch
University of York, GB
- Randy Goebel
University of Alberta, CA
- Valerio Grossi
University of Pisa, IT
- Tias Guns
KU Leuven, BE
- Holger H. Hoos
University of British Columbia – Vancouver, CA
- Frank Hutter
Universität Freiburg, DE
- Kristian Kersting
TU Dortmund, DE
- Lars Kotthoff
University College Cork, IE
- Pauli Miettinen
MPI für Informatik – Saarbrücken, DE
- Mirco Nanni
ISTI-CNR – Pisa, IT
- Benjamin Negrevergne
KU Leuven, BE
- Siegfried Nijssen
KU Leuven, BE
- Barry O'Sullivan
University College Cork, IE
- Andrea Passerini
University of Trento – DISI, IT
- Francesca Rossi
University of Padova, IT
- Lakhdar Sais
Artois University – Lens, FR
- Vijay A. Saraswat
IBM TJ Watson Research Center – Hawthorne, US
- Michele Sebag
University of Paris South XI, FR
- Arno Siebes
Utrecht University, NL
- Guido Tack
Monash Univ. Melbourne, AU
- Yuzuru Tanaka
Hokkaido University, JP
- Joaquin Vanschoren
TU Eindhoven, NL
- Christel Vrain
University of Orleans, FR
- Toby Walsh
NICTA – Sydney, AU