Learning MAX-SAT from Contextual Examples for Combinatorial Optimisation

Mohit Kumar,[1] **Samuel Kolb,**[1] **Stefano Teso,**[2*] **Luc De Raedt**[1]

[1]KU Leuven, [2]University of Trento, Italy

{mohit.kumar, samuel.kolb, luc.deraedt}@cs.kuleuven.be, stefano.teso@unitn.it

Abstract

Combinatorial optimization problems are ubiquitous in artificial intelligence. Designing the underlying models, however, requires substantial expertise, which is a limiting factor in practice. The models typically consist of hard and soft constraints, or combine hard constraints with a preference function. We introduce a novel setting for learning combinatorial optimisation problems from *contextual* examples. These positive and negative examples show – in a particular context – whether the solutions are good enough or not. We develop our framework using the MAX-SAT formalism. We provide learnability results within the realizable and agnostic settings, as well as HASSLE, an implementation based on syntax-guided synthesis and showcase its promise on recovering synthetic and benchmark instances from examples.

Introduction

Combinatorial optimisation is an effective and popular class of techniques for solving real-life problems like scheduling (Demirović, Musliu, and Winter 2019), routing (Mills and Tsang 2000), and planning (Robinson et al. 2010). However, encoding the underlying models often proves to be time-consuming and complicated, as it requires substantial domain and modeling expertise. Therefore, the question arises as to whether such models can be learned from data. This question is studied in constraint learning (Bessiere et al. 2016; De Raedt, Passerini, and Teso 2018), where several algorithms have been developed that automatically acquire theories or mathematical models from examples of past working (positive) and non-working (negative) solutions or analogous forms of supervision.

Combinatorial optimisation models have two components: 1) a set of hard constraints ϕ defining the feasible region, and 2) a preference function f that measures the quality of candidate solutions, sometimes defined as a set of soft constraints. The task of the solver is then to complete a (potentially empty) partial assignment x into a complete assignment xy that is both feasible and optimal, i.e., $xy \models \phi$ and $y \in \operatorname{argmax}_{xy \models \phi} f(xy)$. We use the term *context* to refer

to both partial assignments x and to more general temporary constraints that restrict the outcome of optimisation.

Current learning approaches suffer from two limitations. First, to the best of our knowledge, they do not learn from contextual examples. By doing so, they ignore the fact that the optima can be affected drastically by the context. For instance, $\max_{x:xy \models \phi} f(xy)$ can be very different from $\max_{y:xy \models \phi} f(xy)$. Furthermore, this is also a less realistic setting in practice, as examples of good and bad solutions will always be relative to a context. The reader may notice a resemblance with structured output prediction (e.g. (Tsochantaridis et al. 2005)), where one learns a function f that computes a structured output $y = \operatorname{argmax}_y f(x, y)$ for a given input x. The difference is that in structured prediction the input and output variables are fixed, while in combinatorial optimisation they are not.

Second, existing approaches do not *jointly* learn the hard constraints and the preference function: they either learn one or the other, or else learn them sequentially or independently. But this may break down in applications like personnel rostering. Here, past schedules are often stored in a dataset, but the reasons why a schedule was found to be unacceptable is usually not tracked. In cases like this, a negative example may be either infeasible (because of the hard constraints) or sub-optimal (because of the preference function). This induces a credit-assignment problem that can only be solved by learning the constraints and preference function jointly. See the related work for a more in-depth discussion.

The key contribution of this paper is that we develop a more realistic setting for learning combinatorial optimisation models *from contextual examples* that does not suffer from these limitations. Furthermore, we provide results within this setting for one of the simplest but most fundamental models for combinatorial optimisation, maximum satisfiability (MAX-SAT). Our theoretical analysis shows that MAX-SAT models can be probably approximately correct (PAC) and agnostically learned from contextual data using empirical risk minimization (ERM). Motivated by this, we introduce HASSLE (HArd and Soft conStraint LEarning) an implementation of ERM that exploits syntax-guided synthesis techniques, whereby learning is encoded as an optimization problem and solved using efficient solvers.

Preliminaries

MAX-SAT Let $\boldsymbol{X} = \{X_1, \ldots, X_n\}$ be a set of Boolean variables and $\Phi = \{\phi_1, \ldots, \phi_m\}$ a class of Boolean formulas of interest on $\boldsymbol{X}$, e.g., the set of conjunctions or disjunctions of up to k literals (i.e., variables or their negations). A value assignment $\boldsymbol{x}$ fixes the value of each X_i to x_i.

A *partial maximum satisfiability* (abbreviated MAX-SAT) problem $\mathcal{M}$ is a collection of hard and soft constraints taken from Φ (Miyazaki, Iwama, and Kambayashi 1996; Li and Manya 2009). The hard constraints define the feasible region while the soft ones define a preference relation over feasible configurations. The hard constraints in $\mathcal{M}$ are encoded as a vector $\boldsymbol{c} \in \{0,1\}^m$ such that $c_j = 1$ if ϕ_j appears as a hard constraint in $\mathcal{M}$ and zero otherwise, while the soft constraints as $\boldsymbol{w} \in [-1,1]^m$, where $w_j \neq 0$ if ϕ_j appears as a soft constraint with weight w_j and zero otherwise. This setup allows each $\phi_j \in \Phi$ to be either hard ($c_j = 1$ and w_j arbitrary), soft ($c_j = 0$ and $w_j \neq 0$), or irrelevant ($c_j = w_j = 0$). All MAX-SAT problems can be written in this form by rescaling their weights.

The value $f_{\boldsymbol{w}}(\boldsymbol{x})$ of an assignment $\boldsymbol{x}$ is the total weight of the constraints it satisfies, that is[1]:

$$f_{\boldsymbol{w}}(\boldsymbol{x}) := \sum_{j=1}^{m} w_j \mathbb{1}(\boldsymbol{x} \models \phi_j) \qquad (1)$$

Here, the indicator function $\mathbb{1}(cond)$ evaluates to 1 if $cond$ holds and to 0 otherwise. Inference amounts to finding a feasible configuration that has optimal objective value, that is:

$$\max_{\boldsymbol{x}} \; f_{\boldsymbol{w}}(\boldsymbol{x}) \qquad (2)$$
$$\text{s.t. } (c_j = 1) \implies (\boldsymbol{x} \models \phi_j) \qquad j = 1, \ldots, m$$

where $\boldsymbol{x} \models \phi_j$ indicates that $\boldsymbol{x}$ satisfies ϕ_j. By abbreviating the conjunction of all hard constraints as:

$$\upsilon_{\boldsymbol{c}} := \bigwedge_{j=1}^{m} ((c_j = 1) \Rightarrow \phi_j)$$

we can shorten inference to $\max_{\boldsymbol{x} \models \upsilon_{\boldsymbol{c}}} f_{\boldsymbol{w}}(\boldsymbol{x})$. Although solving MAX-SAT is NP-hard in general (Li and Manya 2009), practical solvers have recently shown impressive performance on highly non-trivial instances[2].

Contexts In real-life applications, decisions are influenced by temporary conditions and restrictions like resource availability. For instance, in personnel scheduling some employees may be unavailable because they are sick or on leave, while in routing tasks part of the network may be down due to a temporary failure. Such restrictions are captured by the notion of *context*. In the previous examples, a context ψ is a conjunction of literals, which fixes the value of one or more decision variables. Letting $\mathcal{M}$ to be a MAX-SAT model, inference in a context ψ amounts to:

$$\max_{\boldsymbol{x} \models \upsilon_{\boldsymbol{c}} \wedge \psi} f_{\boldsymbol{w}}(\boldsymbol{x}) \qquad (3)$$

The set of variables fixed by the context changes too.

Contexts can, however, do more than specifying partial interpretations: in rostering applications an employee may

---

[1]The weights of hard constraints can be safely ignored, as they offset the value of all feasible configurations by the same amount.

[2]See for instance: https://maxsat-evaluations.github.io

be unable to work more than four hours a day due to various contingencies (e.g. pregnancy), while in routing some routes might be taking more than usual due to high congestion. In these cases, the context ψ is an arbitrary Boolean formula that does restrict – but does not necessarily fix – the variables. In the following, Ψ^* will indicate the set of all possible contexts, e.g., the class of all Boolean formulas over $\boldsymbol{X}$ or sub-classes thereof.

Contexts are central in optimization problems as they can significantly alter the quality of optima: optimal schedules may be substantially better if a particularly skilled worker is available. This means ignoring contexts while learning can lead to under-performing models.

Problem Statement

In contrast to existing constraint learning methods, we consider a more realistic setting where example solutions are context-specific. More specifically, we expect each example to be generated as follows: 1) A context $\psi \in \Psi^*$ is observed, 2) A configuration $\boldsymbol{x}$ that satisfies the context ($\boldsymbol{x} \models \psi$) is chosen according to some policy, e.g., by asking a domain expert to provide a good solution, and 3) The configuration is labelled as positive ($y = 1$), if it is feasible and works well enough in practice, and as negative ($y = 0$) otherwise.

This process induces a joint distribution $D(y, \boldsymbol{x}, \psi) = D(y|\boldsymbol{x}, \psi)D(\boldsymbol{x}|\psi)D(\psi)$. We are now ready to define MAX-SAT learning from context-specific data:

Definition 1. *Given a set of Boolean variables $\boldsymbol{X}$, candidate constraints Φ, and context-specific examples S sampled i.i.d. from an unknown joint distribution D, find a MAX-SAT model $\mathcal{M}$ with parameters $\boldsymbol{c}$ and $\boldsymbol{w}$ that can be used to obtain high-quality configurations in any context ψ.*

The exact nature of configuration quality will be formalized in the next section using the notion of regret (Cesa-Bianchi and Lugosi 2006).

A few remarks are in order. First, we do not know which negative examples are infeasible and which are simply not good enough. This introduces an attribution problem: if a negative configuration is infeasible, then $\boldsymbol{c}$ should be responsible for labeling it, otherwise $\boldsymbol{w}$ should. This can be solved elegantly using empirical risk minimization, as shown later on. Second, all configurations are expected to include a description of the context in which they were obtained. This is often the case in real-life applications: in rostering, past schedules are annotated with the unavailable employees, while in routing, the causes of congestion (e.g. maintenance work) are kept track of for monitoring purposes. Applications in which this is not the case can be addressed by treating the unobserved contexts as latent variables. This non-trivial extension is left to future work.

Learnability

In this section, we study MAX-SAT learning from the perspective of statistical learning theory[3]. The analysis is broken into three steps: 1) reducing MAX-SAT learning to

---

[3]As is common in this literature, we focus on statistical aspects and leave computational issues to future work.

classification and showing that the latter is learnable from context-specific examples, 2) showing that classifiers with low context-specific risk correspond to MAX-SAT models with high-quality solutions, and 3) proving that, under mild assumptions, classifiers with low context-specific risk enjoy low risk in general. These results show that it is possible to use ERM to solve MAX-SAT learning from context-specific examples, and motivate our ERM-based approach, HASSLE.

The intra-context case   We consider "MAX-SAT classifiers" of the form:

$$h_{\boldsymbol{c},\boldsymbol{w}}(\boldsymbol{x},\psi) = \mathbb{1}\left(\boldsymbol{x} \in \operatorname*{argmax}_{\boldsymbol{x}' \models \upsilon_{\boldsymbol{c}} \wedge \psi} f_{\boldsymbol{w}}(\boldsymbol{x}')\right)$$

Such classifiers label an instance as positive iff it is feasible with respect to $\upsilon_{\boldsymbol{c}}$ and ψ, *and* optimal with respect to $f_{\boldsymbol{w}}$. Let $\mathcal{H} = \{h_{\boldsymbol{c},\boldsymbol{w}} : \boldsymbol{c} \in \{0,1\}^m, \boldsymbol{w} \in [-1,1]^m\}$ be the set of all MAX-SAT classifiers and $\mathcal{H}_\psi = \{h(\cdot,\psi) : h \in \mathcal{H}\}$ be the set of those with a fixed ψ.

Our analysis relies on the fact that, regardless of the context ψ, $\mathcal{H}_\psi$ has finite Vapnik-Chervonenkis (VC) dimension. In order to show this, we build an easier-to-handle superset $\mathcal{G}_\psi$ of $\mathcal{H}_\psi$ as follows:

$$\mathcal{G}_\psi := \left\{\mathbb{1}\left((\boldsymbol{x} \models \upsilon_{\boldsymbol{c}} \wedge \psi) \wedge f_{\boldsymbol{w}}(\boldsymbol{x}) \ge \max_{\boldsymbol{x}' \models \upsilon_{\boldsymbol{c}'} \wedge \psi} f_{\boldsymbol{w}}(\boldsymbol{x}')\right)\right.$$
$$\left. : \boldsymbol{w} \in [-1,1]^m, \boldsymbol{c}, \boldsymbol{c}' \in \{0,1\}^m \right\}$$

By construction, $VC(\mathcal{H}_\psi) \le VC(\mathcal{G}_\psi)$. The latter can be bounded in terms of two simpler sets $\mathcal{G}_{\psi,\text{hard}}$ and $\mathcal{G}_{\psi,\text{soft}}$:

$$\mathcal{G}_{\psi,\text{hard}} := \{\mathbb{1}(\boldsymbol{x} \models \upsilon_{\boldsymbol{c}} \wedge \psi) : \boldsymbol{c} \in \{0,1\}^m\},$$
$$\mathcal{G}_{\psi,\text{soft}} := \{g_{\boldsymbol{c}',\boldsymbol{w}}(\cdot,\psi) : \boldsymbol{c}' \in \{0,1\}^m, \boldsymbol{w} \in [-1,1]^m\}$$

where:

$$g_{\boldsymbol{c}',\boldsymbol{w}}(\boldsymbol{x},\psi) := \mathbb{1}\left(f_{\boldsymbol{w}}(\boldsymbol{x}) \ge \max_{\boldsymbol{x}' \models \upsilon_{\boldsymbol{c}'} \wedge \psi} f_{\boldsymbol{w}}(\boldsymbol{x}')\right)$$

Lemma 1. *For any ψ and $c \approx 4.7$, it holds that:*
$$VC(\mathcal{G}_\psi) \le c(VC(\mathcal{G}_{\psi,hard}) + VC(\mathcal{G}_{\psi,soft}))$$

Proof. Each and every hypothesis in $\mathcal{G}_\psi$ is the conjunction of a concept from $\mathcal{G}_{\psi,\text{hard}}$ with a concept from $\mathcal{G}_{\psi,\text{soft}}$. The claim follows by applying Theorem 1.1 of (van der Vaart and Wellner 2009). □

The remaining step is to prove that $\mathcal{G}_{\psi,\text{hard}}$ and $\mathcal{G}_{\psi,\text{soft}}$ both have bounded VC dimension. It is clear that $|\mathcal{G}_{\psi,\text{hard}}| \le 2^m$ and therefore $VC(\mathcal{G}_{\psi,\text{hard}}) \le m$. The answer for $\mathcal{G}_{\psi,\text{soft}}$ is given by the following lemma:

Lemma 2. *For any ψ, it holds that $VC(\mathcal{G}_{\psi,soft}) \le m + 1$.*

Proof. $\mathcal{G}_{\psi,\text{soft}}$ can be rewritten as:

$$\mathcal{G}_{\psi,\text{soft}} = \{g_{\boldsymbol{c}',\boldsymbol{w}}(\cdot,\psi) : \boldsymbol{c}' \in \{0,1\}^m, \boldsymbol{w} \in [-1,1]^m\}$$
$$= \{\mathbb{1}(f_{\boldsymbol{w}}(\boldsymbol{x}) \ge \max_{\boldsymbol{x}' \models \upsilon_{\boldsymbol{c}'} \wedge \psi} f_{\boldsymbol{w}}(\boldsymbol{x}')) : \boldsymbol{c}', \boldsymbol{w}\}$$
$$\subset \{\mathbb{1}(f_{\boldsymbol{w}}(\boldsymbol{x}) \ge b) : \boldsymbol{w} \in [-1,1]^m, b \in \mathbb{R}\}$$

Here $\boldsymbol{c}'$ and $\boldsymbol{w}$ implicitly range over $\{0,1\}^m$ and $[-1,1]^m$, respectively. The final set is a subset of $\mathcal{L}$, the class of linear classifiers over the m features $\mathbb{1}(\boldsymbol{x} \models \phi_j(\boldsymbol{x}))$. This shows that $VC(\mathcal{G}_{\text{soft}}) \le VC(\mathcal{L}) \le m + 1$. □

Putting everything together proves that:

Theorem 1. *For every ψ, it holds that $VC(\mathcal{H}_\psi) \le c(2m + 1)$, where c is defined as in Lemma 1.*

As mentioned above, this entails that MAX-SAT classifiers are PAC learnable. Let $L_{D,\psi}(h)$ be the risk of some hypothesis $h \in \mathcal{H}$ applied to some context ψ, i.e.,

$$L_{D,\psi}(h) = \mathbb{E}_{y,\boldsymbol{x}|\psi}\left[\mathbb{1}(h(\boldsymbol{x},\psi) \ne y)\right]$$
$$= \sum_{\boldsymbol{x} \models \psi,\, y} \mathbb{1}(h(\boldsymbol{x},\psi) \ne y)\, D(\boldsymbol{x}, y \mid \psi)$$

Combining Theorem 1 with standard statistical learning results (e.g. Theorem 6.8 in (Shalev-Shwartz and Ben-David 2014)) entails learnability in both the realizable and agnostic settings. In particular, in the realizable setting it holds that:

Corollary 1. *For any $\psi \in \Psi^*$, and $\epsilon_\psi, \delta_\psi \in [0,1]$, there exists an integer $s_\psi(\epsilon_\psi, \delta_\psi) \in \mathbb{N}$ such that, given a dataset S with at least $s_\psi(\epsilon_\psi, \delta_\psi)$ examples specific to ψ, the probability that a hypothesis $h \in \mathcal{H}$ with minimal empirical error on S has risk greater than ϵ_ψ is less than δ_ψ, i.e.:*
$$P(L_{D,\psi}(h) > \epsilon) \le \delta$$

From classification to optimization   We have just shown that ERM can be used to learn low-risk MAX-SAT classifiers for any context. Next, we show that good classifiers correspond to good MAX-SAT models, that is, models useful for combinatorial optimization.

To simplify the presentation, in the remainder of the section, we work in the realizable setting, i.e., there exist a latent model $\mathcal{M}^*$ – or equivalently a hypothesis $h^* \in \mathcal{H}$ – with parameters $\boldsymbol{c}^*, \boldsymbol{w}^*$ and the examples in S are labeled according to it: $y_k = h^*(\boldsymbol{x}_k, \psi_k)$.

Given $h \in \mathcal{H}$, let $\text{opt}_\psi := \{\boldsymbol{x} : \boldsymbol{x} \in \{0,1\}^n, h(\boldsymbol{x},\psi) = 1\}$ be the instances it classifies as positive, i.e., the set of its context-specific optima. Also let opt^*_ψ be the optima of h^* in context ψ. The (average) regret measures the difference in quality between the optima of h and the optima of h^* (Cesa-Bianchi and Lugosi 2006):

Definition 2. *Let $h^* \in \mathcal{H}$ have parameters $\boldsymbol{c}^*, \boldsymbol{w}^*$, and let $\boldsymbol{x}^* \in \text{opt}^*_\psi$. The regret $\text{reg}(\boldsymbol{x},\psi)$ of using configuration $\boldsymbol{x} \models \psi$ in context ψ is $(f_{\boldsymbol{w}^*}(\boldsymbol{x}^*) - f_{\boldsymbol{w}^*}(\boldsymbol{x}))$ if $\boldsymbol{x}$ is feasible with respect to $\upsilon_{\boldsymbol{c}^*}$ or an application-dependent cost $r \in \mathbb{R} \cup \{\infty\}$ otherwise. The average regret $\text{reg}(h,\psi)$ of a hypothesis h in context ψ is the average regret of its optima opt_ψ.*

Next, we show that the average regret of a MAX-SAT model is minimized by optimizing its risk:

Theorem 2. *Let $\alpha_j(\boldsymbol{x}) := \mathbb{1}(c_j = 1 \wedge \boldsymbol{x} \models \phi_j)$ for all $j = 1,\ldots,m$, $\rho = \max_{\boldsymbol{x},\boldsymbol{x}'} \|\boldsymbol{\alpha}(\boldsymbol{x}') - \boldsymbol{\alpha}(\boldsymbol{x})\|_2$, and $\eta = \min_{\boldsymbol{x}:D(\boldsymbol{x}|\psi)>0} D(\boldsymbol{x} \mid \psi)$. For any $h \in \mathcal{H}$ and $\psi \in \Psi^*$, it holds that:*

$$\text{reg}(h,\psi) \le \frac{\|\boldsymbol{w}^*\|_2\, \rho + r}{|\text{opt}_\psi|\eta} L_{D,\psi}(h)$$

Proof. First, notice that:

$$\sum_{\boldsymbol{x}:h(\boldsymbol{x},\psi)=1} \mathbb{1}(h(\boldsymbol{x},\psi) \ne h^*(\boldsymbol{x},\psi)) \le \frac{1}{\eta} L_{D,\psi}(h) \quad (4)$$

Indeed, the risk can be lower bounded as:

$$L_{D,\psi}(h) = \sum_{\boldsymbol{x}\in\{0,1\}^n} \mathbb{1}(h(\boldsymbol{x},\psi) \neq h^*(\boldsymbol{x},\psi))\, D(\boldsymbol{x}\,|\,\psi)$$

$$\geq \sum_{\boldsymbol{x}\,:\,h(\boldsymbol{x},\psi)=1} \mathbb{1}(h^*(\boldsymbol{x},\psi)=0)\, D(\boldsymbol{x}\,|\,\psi)$$

$$\geq \eta \sum_{\boldsymbol{x}\,:\,h(\boldsymbol{x},\psi)=1} \mathbb{1}(h^*(\boldsymbol{x},\psi)=0)$$

Second, notice that $\boldsymbol{x}$ contributes to the average regret iff $h(\boldsymbol{x},\psi) = 1 \wedge h^*(\boldsymbol{x},\psi) = 0$: if $\boldsymbol{x}$ is feasible w.r.t. $\upsilon_{\boldsymbol{c}^*}$, it contributes a factor $(f_{\boldsymbol{w}^*}(\boldsymbol{x}^*) - f_{\boldsymbol{w}^*}(\boldsymbol{x}))$, else it contributes r. Thus, ignoring constants, the regret can be written as the sum of two terms. The first one is:

$$\sum_{\boldsymbol{x}\,:\,h(\boldsymbol{x},\psi)=1} \mathbb{1}(h^*(\boldsymbol{x},\psi)=0 \wedge \boldsymbol{x} \models \upsilon_{\boldsymbol{c}^*})\,(f_{\boldsymbol{w}^*}(\boldsymbol{x}^*) - f_{\boldsymbol{w}^*}(\boldsymbol{x}))$$

which is upper bounded by:

$$\left[\max_{\boldsymbol{x},\boldsymbol{x}'} f_{\boldsymbol{w}^*}(\boldsymbol{x}) - f_{\boldsymbol{w}^*}(\boldsymbol{x}')\right] \sum_{\boldsymbol{x}\,:\,h(\boldsymbol{x},\psi)=1} \mathbb{1}(h^*(\boldsymbol{x},\psi)=0 \wedge \boldsymbol{x} \models \upsilon_{\boldsymbol{c}^*})$$

By Cauchy-Schwarz, the left factor is at most $\|\boldsymbol{w}^*\|_2\, \rho$. The second term is:

$$r \sum_{\boldsymbol{x}\,:\,h(\boldsymbol{x},\psi)=1} \mathbb{1}(h^*(\boldsymbol{x},\psi)=0 \wedge \boldsymbol{x} \not\models \upsilon_{\boldsymbol{c}^*})$$

The summations in both are upper bounded by the left hand side of Eq (4). Dividing their sum by $|\mathrm{opt}_\psi|$ gives the average regret and proves the claim. $\square$

In many real-life applications, the cost r of obtaining an infeasible configuration is finite. This is not true in applications where infeasible configurations violate societal or safety requirements, where $r = \infty$. In this case, the bound in Theorem 2 becomes loose, unless the risk is exactly zero. This shows that, understandably, extra care should be taken when applying machine learning to high-stakes domains.

Generalization to the context-independent case So far we have only considered learning and inference within the same context. However, learned models are likely going to be used in previously unobserved contexts. Next, we show that MAX-SAT classifiers learned from context-specific data do generalize, under mild assumptions, to the top context – and therefore have low regret in the top context too. Generalization to *arbitrary* contexts can be ensured using similar arguments. Since this complicates exposition, we postpone a more complete discussion to a longer version of the paper.

Intuitively, generalization requires the observed contexts to be collectively *representative* of the global one. More formally, we say that Ψ is representative if whenever $h(\cdot,\top)$ makes a mistake, there is at least a context in Ψ that "catches" that mistake:

Definition 3. *For a given true hypothesis h^*, a collection of contexts Ψ is* representative *if and only if for every h and $\boldsymbol{x}$ it holds that:*

$$(h(\boldsymbol{x},\top) \neq h^*(\boldsymbol{x},\top)) \implies$$
$$\bigvee_{\psi\in\Psi}(\boldsymbol{x} \models \psi \wedge h(\boldsymbol{x},\psi) \neq h^*(\boldsymbol{x},\psi)) \quad (5)$$

Figure 1: Example representative contexts: $\psi_1 = X_1$ (blue), $\psi_2 = \neg X_1$ (violet), $\psi_3 = X_2 \wedge X_3$ (red) are representative for $\Phi = \{X_1, X_2, X_3, \neg X_1, \neg X_2, \neg X_3\}$ and $\boldsymbol{w}^* = \{1, 1, 1, 0, 0, 0\}$; details in the text. (Best viewed in color.)

To see why representativeness is necessary, consider a ground truth model $\boldsymbol{c}^* = 0$ and $\boldsymbol{w}^* = (1, 1, 1, 0, 0, 0)$ with:

$$\Phi = \{X_1, X_2, X_3, \neg X_1, \neg X_2, \neg X_3\}$$
$$\psi_1 = X_1, \quad \psi_2 = \neg X_1, \quad \psi_3 = X_2 \wedge X_3$$

Notice that under ψ_1 the optimum is $(1, 1, 1)$ with value 3, while under ψ_2 the optimum is $(0, 1, 1)$ with value 2. Now, take a hypothesis h that misclassifies $\boldsymbol{x} = (0, 1, 1)$ as a global optimum. The error does not show up in either ψ_1 (because $\boldsymbol{x}$ does not lie in it) nor in ψ_2 (because $\boldsymbol{x}$ is indeed optimal in it). Therefore, a classifier with arbitrarily low risk on both ψ_1 and ψ_2 may still misclassify $\boldsymbol{x}$. Requiring that h performs well also on ψ_3 fixes this issue, because $\boldsymbol{x}$ does lie in ψ_3 but is not optimal in it, cf. Figure 1.

Crucially, if h has low risk on all observed contexts Ψ and Ψ is representative, then it performs well in general too:

Lemma 3. *If Ψ is representative and for every context $\psi \in \Psi$ it holds that $\forall \boldsymbol{x} \models \psi\,.\,D(\boldsymbol{x}\,|\,\psi) > 0$, there is a finite constant $\beta \geq 0$ such that for every $h \in \mathcal{H}$:*

$$L_{D,\top}(h) \leq \beta \sum_{\psi\in\Psi} L_{D,\psi}(h)$$

Proof. Representativeness of Ψ entails:

$$\mathbb{1}\{h(\boldsymbol{x},\top) \neq h^*(\boldsymbol{x},\top)\}$$
$$\leq \mathbb{1}\left(\bigvee_{\psi\in\Psi}(\boldsymbol{x} \models \psi \wedge h(\boldsymbol{x},\psi) \neq h^*(\boldsymbol{x},\psi))\right)$$
$$\leq \sum_{\psi\in\Psi} \mathbb{1}(\boldsymbol{x} \models \psi \wedge h(\boldsymbol{x},\psi) \neq h^*(\boldsymbol{x},\psi))$$

Plugging this into the risk of h in context $\top$ gives:

$$\sum_{\boldsymbol{x}\in\{0,1\}^n} \mathbb{1}(h(\boldsymbol{x},\top) \neq h^*(\boldsymbol{x},\top))\, D(\boldsymbol{x}\,|\,\top)$$
$$\leq \sum_{\boldsymbol{x}} \sum_{\psi\in\Psi} \mathbb{1}(\boldsymbol{x} \models \psi \wedge h(\boldsymbol{x},\psi) \neq h^*(\boldsymbol{x},\psi))\, D(\boldsymbol{x}\,|\,\top)$$
$$= \sum_{\psi\in\Psi} \sum_{\boldsymbol{x}\models\psi} \mathbb{1}(h(\boldsymbol{x},\psi) \neq h^*(\boldsymbol{x},\psi))\, D(\boldsymbol{x}\,|\,\top)$$

Letting $\beta_\psi = \max_{\boldsymbol{x}\models\psi} \frac{D(\boldsymbol{x}\,|\,\top)}{D(\boldsymbol{x}\,|\,\psi)}$, we can write:

$$L_{D,\top}(h) \leq \sum_{\psi\in\Psi} \beta_\psi \sum_{\boldsymbol{x}\models\psi} \mathbb{1}(h(\boldsymbol{x},\psi) \neq h^*(\boldsymbol{x},\psi))\, D(\boldsymbol{x}\,|\,\psi)$$
$$= \sum_{\psi\in\Psi} \beta_\psi L_{D,\psi}(h) \leq \beta \sum_{\psi\in\Psi} L_{D,\psi}(h)$$

where we chose $\beta = \max_{\psi \in \Psi} \beta_\psi$. It only remains to show that β is finite. Notice that for each β_ψ the max runs over those $\boldsymbol{x}$'s that satisfy ψ, hence $D(\boldsymbol{x}|\psi)$ is always non-zero, and so each β_ψ is finite. This proves that β is finite as it is maximum over a set of finite value elements. $\qquad\square$

We are now ready to state our main result:

Theorem 3. *In the realizable case, for any Ψ and D that satisfy the conditions of Lemma 3 and for every ϵ, $\delta \in (0,1)$, there exist integers $t_\psi(\epsilon, \delta)$ for $\psi \in \Psi$ such that, if S contains at least $t_\psi(\epsilon, \delta)$ context-specific examples for every $\psi \in \Psi$, then any hypothesis h with minimal empirical risk on S satisfies:*

$$P(L_{D,\top}(h) > \epsilon) < \delta$$

Proof. In the realizable case, any empirical risk minimizer h attains zero empirical risk over S:

$$\sum_{(\boldsymbol{x},\psi) \in S} \mathbb{1}(h(\boldsymbol{x},\psi) \neq h^*(\boldsymbol{x},\psi)) = 0$$

and therefore, since all terms appearing in the sum are non-negative, $L_{S,\psi}(h) = 0$ for all $\psi \in \Psi$. In turn, Corollary 1 guarantees that for every ϵ_ψ, $\delta_\psi \in (0,1)$, there exists an integer $s_\psi(\epsilon_\psi, \delta_\psi)$ such that if S includes at least $s_\psi(\epsilon_\psi, \delta_\psi)$ examples for context ψ, then:

$$P(L_{D,\psi}(h) > \epsilon_\psi) < \delta_\psi \qquad\qquad \psi \in \Psi \qquad (6)$$

The second part of the proof involves upper bounding the probability that h has large risk in context $\top$:

$$
\begin{aligned}
P(L_{D,\top}(h) \geq \epsilon) &\leq P(\beta \textstyle\sum_{\psi \in \Psi} L_{D,\psi}(h) \geq \epsilon) \\
&\leq P(\beta |\Psi| \max_{\psi \in \Psi} L_{D,\psi}(h) \geq \epsilon) \\
&= P(\max_{\psi \in \Psi} L_{D,\psi}(h) \geq \tfrac{\epsilon}{\beta|\Psi|}) \\
&\leq P(\textstyle\bigvee_{\psi \in \Psi}(L_{D,\psi}(h) \geq \tfrac{\epsilon}{\beta|\Psi|})) \\
&\leq \textstyle\sum_{\psi \in \Psi} P(L_{D,\psi}(h) \geq \tfrac{\epsilon}{\beta|\Psi|}) \qquad (7)
\end{aligned}
$$

The first step follows from Lemma 3. By Eq (6), the last expression can be made smaller than any δ by adding enough context-specific examples to S. In particular, having at least:

$$t_\psi(\epsilon, \delta) = s_\psi\left(\frac{\epsilon}{\beta|\Psi|}, \delta_\psi\right)$$

examples for every context guarantees that Eq (7) is less than any $\delta = \sum_{\psi \in \Psi} \delta_\psi$. This concludes the proof. $\qquad\square$

Summarizing, so long as the observed contexts are representative and there are enough examples, ERM learns a low-risk MAX-SAT model that has low regret in both the observed and in the global context with high probability.

Remarks Our results can be easily extended to more general settings: 1) Nothing in our theory relies on the variables being Boolean nor on the constraints being propositional formulas. Hence, our results transfer to general weighted constraint satisfaction problems with discrete variables and additive preference functions (Rossi and Sperduti 2004; Schiex et al. 1995). 2) Theorem 2 can be adapted to the case where positive examples are high-quality yet not optimal. This is often the case in practice, as examples are likely produced by a human domain expert. 3) Theorem 3 can be easily adapted to the agnostic case by using standard generalization bounds based on the VC dimension (Anthony and Bartlett 2009). 4) The bound in Lemma 3 can be made tighter by careful tracking of *which* contexts in Ψ catch an error for a particular h and $\boldsymbol{x}$. In order to keep the presentation accessible, we will detail these extensions in a longer version of the paper.

MAX-SAT Learning with HASSLE

We are finally ready to present HASSLE, a prototype implementation of our ERM-based algorithm. Given $\boldsymbol{X}$, Φ, and a context-specific dataset S encompassing contexts Ψ, finding a MAX-SAT model $\mathcal{M}$ with minimal empirical error equates to solving the following optimization problem:

$$\text{find } \boldsymbol{c} \in \{0,1\}^m, \boldsymbol{w} \in [-1,1]^m$$

$$\text{s.t. } y_k \Leftrightarrow \left(\boldsymbol{x}_k \in \operatorname*{argmax}_{\boldsymbol{x}' \models \upsilon_{\boldsymbol{c}} \wedge \psi_k} f_{\boldsymbol{w}}(\boldsymbol{x}') \right) \quad k = 1, \dots, s \quad (8)$$

Following the literature on syntax-guided synthesis (Alur et al. 2018), we would like to solve this problem using an appropriate solver. However, this is problematic, as Eq (8) implicitly defines s nested partial MAX-SAT problems.

We circumvent this difficulty by converting the above into a mixed-integer linear program (MILP), shown in Figure 2. The encoding can be split into two parts. The objective function (Eq (9)) maximizes a set of per-context variables $\gamma_\psi \in \mathbb{R}$ and together with Eq (10) computes a feasible optimum $\boldsymbol{x}'_{\psi_\ell}$ for every context $\psi_\ell \in \Psi$ in the data. Next, Eq (11) ensures that $\boldsymbol{c}$ and $\boldsymbol{w}$ are chosen so that every positive example $\boldsymbol{x}_k$ is feasible and optimal in its own context ψ_k – using the context-specific optimum $\boldsymbol{x}'_{\psi_k}$ as a reference – and that no negative instance is. Once solved, the learned MAX-SAT model is read off from $\boldsymbol{c}$ and $\boldsymbol{w}$. Two important remarks are in order. First, γ_ψ is always bounded thanks to Eq (10). For contexts that have at least one positive example, Eq (10) is superfluous and can be omitted for efficiency. Second, the MILP encoding does not exactly correspond to the encoding in Eq (8) – it is, instead, a tight approximation.

To solve the nested MAX-SAT problems, the MILP encoding uses variables γ_{ψ_ℓ} to represent the value of an optimal solution in a context ψ_ℓ according to the MAX-SAT problem encoded by $\boldsymbol{c}$ and $\boldsymbol{w}$. However, instead of individually maximizing the γ_{ψ_ℓ}, it has to resort to maximizing the sum $\sum_{\psi_\ell} \gamma_{\psi_\ell}$ instead. This summation introduces a dependency that allows the encoding to choose γ suboptimally for some contexts – allowing it to falsely label examples as positive that are not positive w.r.t. to the current $\boldsymbol{c}$ and $\boldsymbol{w}$ – in order to increase the γ's for different contexts (an example of this behavior is included in the supplementary material, Example 1). A simple post-processing step can detect such suboptimal γ's and an iterative procedure[4] could be used to solve the problem, but our experimental results show that

---

[4]In cases where falsely labeled positives are detected, an iterative procedure could be used to solve the problem to optimality: For

$$\max_{\boldsymbol{c},\boldsymbol{w},\boldsymbol{\gamma}} \ \sum_{\psi_\ell \in \Psi} \gamma_{\psi_\ell} \tag{9}$$

$$\text{s.t.} \ (\boldsymbol{x}'_{\psi_\ell} \models \upsilon_{\boldsymbol{c}} \wedge \psi_\ell) \wedge (\gamma_{\psi_\ell} \le f_{\boldsymbol{w}}(\boldsymbol{x}'_{\psi_\ell})) \qquad\qquad \psi_\ell \in \Psi \tag{10}$$

$$y_k \Leftrightarrow (\boldsymbol{x}_k \models \upsilon_{\boldsymbol{c}} \wedge \psi_k) \wedge (\gamma_{\psi_k} \le f_{\boldsymbol{w}}(\boldsymbol{x}_k)) \qquad\qquad k = 1, \dots, s \tag{11}$$

$$\boldsymbol{\gamma} \in \mathbb{R}^{|\Psi|}, \ \boldsymbol{c} \in \{0,1\}^m, \ \boldsymbol{w} \in [0,1]^m$$

Figure 2: Simplified MILP encoding of HASSLE. The full encoding is given in the Supplement due to space restrictions.

this mismatch occurs rarely in practice. Indeed, in our experiments, the MILP encoding makes a mistake on the training set less than 1% of the time.

Experiments

We answer the following research questions: **Q1)** Does HASSLE acquire accurate, low-regret models? **Q2)** Do more examples lead to a better model? **Q3)** How well does it scale to more complex target models?

To this end, we used our MILP encoding for recovering synthetic and benchmark MAX-SAT models $\mathcal{M}^*$ of increasing complexity from examples. The experimental setup can be found at https://github.com/mohitKULeuven/hassle.

Datasets For the experiments we used both synthetic and benchmark models. The synthetic models were generated by enumerating all possible disjunctions of up to k literals of n variables, and then sampling $\boldsymbol{c}^*$ and $\boldsymbol{w}^*$ at random so that they have exactly m_{hard} and m_{soft} non-overlapping non-zero entries, respectively, and the weights are sampled from $[-1, 1]$ uniformly. A total of 36 configurations were obtained by varying $n = 5, 10, 15$, $k = 2, 5$ $m_{\text{hard}} = 0, 5, 10$, and $m_{\text{soft}} = 5, 10$, and 5 random models were synthesized for every configuration, giving a total of 180 models.

Benchmark MAX-SAT instances are typically huge, with thousands of variables and constraints. So, to make sure that HASSLE can learn hard combinatorial problems, we instead chose five phase-transition SAT instances (Gent and Walsh 1994) with 20 variables and 91 constraints [5], and for each of them, turned most of the hard constraints into soft ones by adding random weights.

A dataset S was collected for each model $\mathcal{M}^*$ by sampling $|\Psi| = 2, 5, 10$ random contexts (that are not the $\top$ context) and then taking s^+ positive and s^- negative examples for each context. However when s^+ is large, for some contexts there may not exist s^+ positive examples. For these cases, we used as many as possible and in the results we report the average number $\hat{s}^+$. The contexts were chosen to be conjunctions of literals, which act like partial assignments.

every context ψ_ℓ for which there exists an instance $\boldsymbol{x}'$ with a value larger than a positive instance $\boldsymbol{x}^+ \models \psi_\ell$: $f_{\boldsymbol{w}}(\boldsymbol{x}') > f_{\boldsymbol{w}}(\boldsymbol{x}^+)$, a constraint: $\boldsymbol{x}' \not\models \upsilon_{\boldsymbol{c}} \vee f_{\boldsymbol{w}}(\boldsymbol{x}') \le f_{\boldsymbol{w}}(\boldsymbol{x}^+)$ is added to the encoding. The extended encoding is solved and this procedure is repeated until all instances are labeled correctly.

[5]Taken from: www.cs.ubc.ca/~hoos/SATLIB/benchm.html

Performance measures The performance of learned models $\mathcal{M}$ was captured using precision and average regret. Precision measures how often the optimal solutions of $\mathcal{M}$ are feasible in $\mathcal{M}^*$, while regret measures how far the optimal solutions in $\mathcal{M}$ are from the ones in $\mathcal{M}^*$. To calculate precision, first we generate $100 \times 5^{\frac{n}{10}}$ optimal examples using $\mathcal{M}$ and then calculate the percentage of feasible ones. Increasing n leads to increase in the set of optimal examples, that is why we increase the sample size with n. Next we compute the regret for the set of feasible examples using definition 2. To simplify the comparison, the regret is normalized to $[0, 1]$ by dividing it by $f_{\boldsymbol{w}^*}(\boldsymbol{x}^*)$, where $\boldsymbol{x}^*$ is an optimum of $\mathcal{M}^*$. In order to assess generalization across contexts, precision and regret are computed in both the global context $\top$ and random contexts not used in training. GUROBI was used to solve the MILP encoding on an Intel(R) Xeon(R) CPU E3-1225 v3 @ 3.20GHz with 32 GiB RAM.

Results on synthetic models To answer **Q1**, we evaluated HASSLE on ground truth models with an increasing number of hard constraints m_{hard} while keeping the number of positive and negative examples fixed (to $\hat{s}^+ = 2.4$ and $s^- = 5$). The other parameters are averaged over. The results are reported in Figure 3 (left). The two left bar plots show the results for the global context, while the right ones the results for the random contexts. Increasing the number of hard constraints understandably leads to a drop in precision and a slight increase in regret, although the latter is less than 10% on average. Learning also takes less than $5s$ per model.

We also looked into the effect of the number of contexts by increasing $|\Psi| = 2, 5, 10$: precision does increase slightly (around $+4\%$) while regret remains stable in both global and random contexts. This may be because the training contexts have a non-trivial overlap and have common optima.

An important part of the experiment was to see if the performance can be improved by increasing s^+, s^- as claimed in the theory. So to answer **Q2** we did experiments by fixing some parameters ($n = 10$, $k = 2$, $m_{\text{hard}} = 5$, $m_{\text{soft}} = 5$, $|\Psi| = 20$) and varying the number of examples (s^+, s^-) used to learn the model, see the right graph in Figure 3. As can be noticed, increasing the number of examples improves the precision as well as recall, although as expected the time taken also increases fast (from 4 Seconds to 6 hours). We remark that HASSLE is a proof-of-concept implementation, and that it can be sped-up by optimizing the encoding and using techniques like explicit cutting planes (Joachims, Fin-
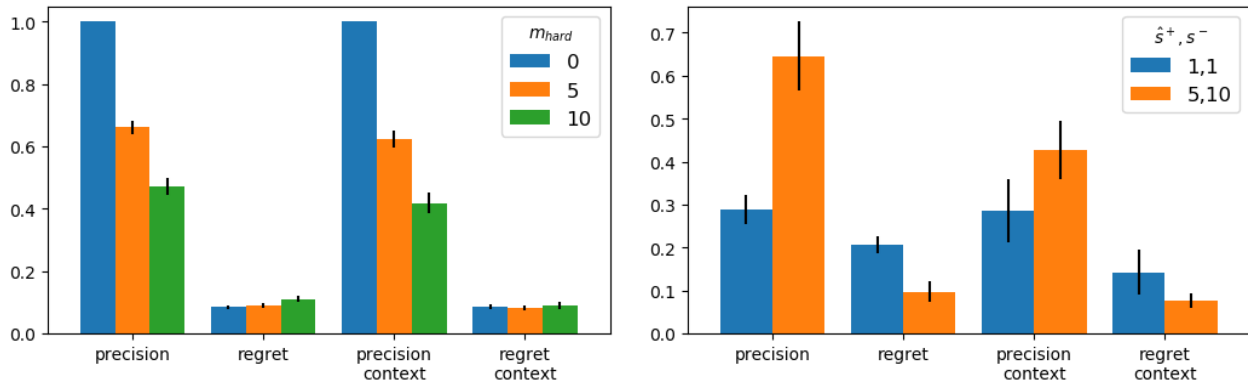
Figure 3: Left: Performance versus number of hard constraints in $\mathcal{M}^*$. Right: Performance versus number of examples used for learning. Increasing the number of hard constraints in $\mathcal{M}^*$ decreases the precision, although does not impact regret much. However increasing the number of examples improves the model by both increasing the precision and decreasing the regret.

ley, and Yu 2009), symmetry-breaking constraints and exact incremental learning (Kolb et al. 2018). A detailed exploration of this is left to future work. Notice also that, once learnt, these models can be used multiple times on different contexts, justifying the cost of learning. So **Q3** can be answered affirmatively. Summarizing, HASSLE almost always produces a low regret model, with more complex models requiring more examples to be learned at the cost of run-time.

Results on benchmark models The experiments were run on 5 benchmark instances, fixing $|\Psi| = 2$, $s = 50$, $m_{\text{hard}} = 6$ and $m_{\text{soft}} = 85$. The average precision observed was 91.7% in the global context, while it drops to 73.5% for randomly generated contexts. However the average regret remains $\approx 5\%$ in both the cases.

Related Work

HASSLE is closely related to constraint learning and acquisition (Bessiere et al. 2016; De Raedt, Passerini, and Teso 2018). There contexts, which are pervasive in real-life decision making, are usually ignored. One exception is QuAcq (Bessiere et al. 2013), which acquires hard constraints from membership queries about partial assignments. QuAcq, however, is allowed to *choose* informative partial assignments, while in our (harder) case the contexts are fixed by the environment. Further, QuAcq has neither support for preference functions nor for general contexts.

Most approaches in constraint learning acquire hard constraints only by searching the version space with one-directional (Beldiceanu and Simonis 2012) or bi-directional search (Bessiere et al. 2016). HASSLE leverages ideas from syntax-guided synthesis (Alur et al. 2018), where learning is cast as a proper satisfaction or optimization problem and tackled with a solver. This strategy has been used for learning rules (Malioutov and Meel 2018), hybrid logical-numerical formulas (Kolb et al. 2018), Bayesian networks (Berg, Järvisalo, and Malone 2014), and causal models (Hyttinen et al. 2013).

Approaches that learn soft constraints build on machine

learning techniques such as regression (Rossi and Sperduti 2004), learning-to-rank (Pawlak and Krawiec 2017), or structured prediction (Teso, Sebastiani, and Passerini 2017), depending on the kind of available supervision. However, these methods require the hard constraints to be given.

Other related areas are structured prediction (McAllester 2007; London, Huang, and Getoor 2016) and contextual (combinatorial) bandits (Li et al. 2010). In these settings the supervision consists of input-output pairs for which the partitioning of variables into inputs and outputs is fixed. This schema is much more restrictive than general context-specific examples. In addition, in structured prediction there is no support for learning hard constraints.

MPE inference in Bayesian networks and Markov Logic Networks can be cast as weighted MAX-SAT (Park 2002; Richardson and Domingos 2006). However, MLNs are not learned from contextual examples where both context x and completion xy are provided. Rather the xy are sampled from the underlying probability distribution and it is not assumed that the examples are optimal (i.e., maximally likely).

Conclusion

We introduced the novel learning task of acquiring combinatorial optimization models from contextual data, focusing specifically on MAX-SAT models. Our analysis shows that empirical risk minimization (ERM) provably learns low-regret MAX-SAT models from context-specific examples, and justifies our ERM-based implementation, HASSLE. The latter relies on an approximate but tight MILP encoding to perform learning and, given enough examples, is able to recover both synthetic and benchmark instances.

References

[Alur et al. 2018] Alur, R.; Singh, R.; Fisman, D.; and Solar-Lezama, A. 2018. Search-based program synthesis. *Communications of the ACM* 61(12):84–93.

[Anthony and Bartlett 2009] Anthony, M., and Bartlett, P. L. 2009. *Neural network learning: Theoretical foundations.* cambridge university press.

[Beldiceanu and Simonis 2012] Beldiceanu, N., and Simonis, H. 2012. A Model Seeker: Extracting global constraint models from positive examples. In *International Conference on Principles and Practice of Constraint Programming*, 141–157. Springer.

[Berg, Järvisalo, and Malone 2014] Berg, J.; Järvisalo, M.; and Malone, B. 2014. Learning optimal bounded treewidth bayesian networks via maximum satisfiability. In *Artificial Intelligence and Statistics*, 86–95.

[Bessiere et al. 2013] Bessiere, C.; Coletta, R.; Hebrard, E.; Katsirelos, G.; Lazaar, N.; Narodytska, N.; Quimper, C.-G.; and Walsh, T. 2013. Constraint acquisition via partial queries. In *Twenty-Third International Joint Conference on Artificial Intelligence*.

[Bessiere et al. 2016] Bessiere, C.; Daoudi, A.; Hebrard, E.; Katsirelos, G.; Lazaar, N.; Mechqrane, Y.; Narodytska, N.; Quimper, C.-G.; and Walsh, T. 2016. New approaches to constraint acquisition. In *Data mining and constraint programming*. Springer. 51–76.

[Cesa-Bianchi and Lugosi 2006] Cesa-Bianchi, N., and Lugosi, G. 2006. *Prediction, learning, and games*. Cambridge university press.

[De Raedt, Passerini, and Teso 2018] De Raedt, L.; Passerini, A.; and Teso, S. 2018. Learning constraints from examples. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

[Demirović, Musliu, and Winter 2019] Demirović, E.; Musliu, N.; and Winter, F. 2019. Modeling and solving staff scheduling with partial weighted MaxSAT. *Annals of Operations Research* 275(1):79–99.

[Gent and Walsh 1994] Gent, I. P., and Walsh, T. 1994. The sat phase transition. In *ECAI*.

[Hyttinen et al. 2013] Hyttinen, A.; Hoyer, P. O.; Eberhardt, F.; and Järvisalo, M. 2013. Discovering cyclic causal models with latent variables: a general sat-based procedure. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence*, 301–310. AUAI Press.

[Joachims, Finley, and Yu 2009] Joachims, T.; Finley, T.; and Yu, C.-N. J. 2009. Cutting-plane training of structural SVMs. *Machine Learning* 77(1):27–59.

[Kolb et al. 2018] Kolb, S.; Teso, S.; Passerini, A.; and De Raedt, L. 2018. Learning SMT(LRA) Constraints using SMT Solvers. In *IJCAI*, 2333–2340.

[Li and Manya 2009] Li, C. M., and Manya, F. 2009. MaxSAT, Hard and Soft Constraints. *Handbook of satisfiability* 185:613–631.

[Li et al. 2010] Li, L.; Chu, W.; Langford, J.; and Schapire, R. E. 2010. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, 661–670. ACM.

[London, Huang, and Getoor 2016] London, B.; Huang, B.; and Getoor, L. 2016. Stability and generalization in structured prediction. *The Journal of Machine Learning Research* 17(1):7808–7859.

[Malioutov and Meel 2018] Malioutov, D., and Meel, K. S. 2018. MLIC: A MaxSAT-Based framework for learning interpretable classification rules. In *International Conference on Principles and Practice of Constraint Programming*, 312–327. Springer.

[McAllester 2007] McAllester, D. 2007. Generalization bounds and consistency. *Predicting structured data* 247–261.

[Mills and Tsang 2000] Mills, P., and Tsang, E. 2000. Guided local search for solving sat and weighted max-sat problems. *Journal of Automated Reasoning* 24(1-2):205–223.

[Miyazaki, Iwama, and Kambayashi 1996] Miyazaki, S.; Iwama, K.; and Kambayashi, Y. 1996. Database queries as combinatorial optimization problems. In *Proceedings of the International Symposium on Cooperative Database Systems for Advanced Applications*, 448–454.

[Park 2002] Park, J. D. 2002. Using weighted max-sat engines to solve mpe. In *AAAI/IAAI*, 682–687.

[Pawlak and Krawiec 2017] Pawlak, T. P., and Krawiec, K. 2017. Automatic synthesis of constraints from examples using mixed integer linear programming. *European Journal of Operational Research* 261(3):1141–1157.

[Richardson and Domingos 2006] Richardson, M., and Domingos, P. 2006. Markov logic networks. *Machine learning* 62(1-2):107–136.

[Robinson et al. 2010] Robinson, N.; Gretton, C.; Pham, D. N.; and Sattar, A. 2010. Partial weighted maxsat for optimal planning. In *Pacific rim international conference on artificial intelligence*, 231–243. Springer.

[Rossi and Sperduti 2004] Rossi, F., and Sperduti, A. 2004. Acquiring both constraint and solution preferences in interactive constraint systems. *Constraints* 9(4):311–332.

[Schiex et al. 1995] Schiex, T.; Fargier, H.; Verfaillie, G.; et al. 1995. Valued constraint satisfaction problems: Hard and easy problems. *IJCAI (1)* 95:631–639.

[Shalev-Shwartz and Ben-David 2014] Shalev-Shwartz, S., and Ben-David, S. 2014. *Understanding machine learning: From theory to algorithms*. Cambridge university press.

[Teso, Sebastiani, and Passerini 2017] Teso, S.; Sebastiani, R.; and Passerini, A. 2017. Structured learning modulo theories. *Artificial Intelligence* 244:166–187.

[Tsochantaridis et al. 2005] Tsochantaridis, I.; Joachims, T.; Hofmann, T.; and Altun, Y. 2005. Large margin methods for structured and interdependent output variables. *Journal of machine learning research* 6(Sep):1453–1484.

[van der Vaart and Wellner 2009] van der Vaart, A., and Wellner, J. 2009. A note on bounds for VC dimensions. *Institute of Mathematical Statistics collections* 5:103–107.