Graphical Models – propositional logic and probabilistic reasoning

LAAS/CNRS confined seminar

M.C. Cooper[1], S. de Givry[2] T. Schiex[2] & C. Brouard[2] (learning)

[1] Université Fédérale de Toulouse, ANITI, IRIT, Toulouse, France
[2] Université Fédérale de Toulouse, ANITI, INRAE MIAT, UR 875, Toulouse, France

More details in the STACS'2020 tutorial

May 5, 2020

ANITI
ARTIFICIAL & NATURAL INTELLIGENCE TOULOUSE INSTITUTE

INRAE
science for people, life & earth

Description of a multivariate function as the combination of simple functions

- discrete models: the function takes discrete variables as inputs
- we stick to totally ordered co-domains (non negative, optimization)
- combination: through a (well-behaved) binary operator

What is a graphical model?

Description of a multivariate function as the combination of simple functions

- discrete models: the function takes discrete variables as inputs
- we stick to totally ordered co-domains (non negative, optimization)
- combination: through a (well-behaved) binary operator

What functions?

- Boolean functions: propositional logical reasoning
- Numerical functions (integer, real): reasoning with cost or probabilities
- infinite valued or bounded functions: logic (feasibility) + cost/probabilities

System modeling for optimization, analysis, design…

- The function describes a system property
- Explore it: find its minimum (feasibility, optimisation), or average value (counting)

System modeling for optimization, analysis, design…

- The function describes a system property
- Explore it: find its minimum (feasibility, optimisation), or average value (counting)

Example

A digital circuit	value of the output
A schedule or a time-table	feasibility, acceptability
A pedigree with partial genotypes	Mendel consistency, probability
A frequency assignment	interference amount
A 3D molecule	energy, stability

What for?

System modeling for optimization, analysis, design…

- The function describes a system property
- Explore it: find its minimum (feasibility, optimisation), or average value (counting)

Example

- A digital circuit — value of the output
- A schedule or a time-table — feasibility, acceptability
- A pedigree with partial genotypes — Mendel consistency, probability
- A frequency assignment — interference amount
- A 3D molecule — energy, stability

Computationally hard

concise description of a multi-dimensional object, little properties

Definition (Graphical Model (GM))

A GM $\mathcal{M} = \langle \boldsymbol{V}, \Phi \rangle$ with co-domain B and combination operator $\oplus$ is defined by:

- a sequence of n variables $\boldsymbol{V}$, each with an associated finite domain of size less than d.
- a set Φ of e functions (or factors).
- Each function $\varphi_{\boldsymbol{S}} \in \Phi$ is a function from $D^{\boldsymbol{S}} \rightarrow B$. $\boldsymbol{S}$ is called the scope of the function and $|\boldsymbol{S}|$ its arity.

Definition (Joint function)

$$\Phi_{\mathcal{M}}(\boldsymbol{v}) = \bigoplus_{\varphi_{\boldsymbol{S}} \in \Phi} \varphi_{\boldsymbol{S}}(\boldsymbol{v}[\boldsymbol{S}])$$

Definition (Constraint network (used in Constraint programming))

A GM $\mathcal{M} = \langle \boldsymbol{V}, \Phi \rangle$ defined by:

- a sequence of n variables $\boldsymbol{V}$, each with an associated finite domain of size less than d.
- a set Φ of e Boolean functions (or constraints).

Definition (Joint function)

$$\Phi_{\mathcal{M}}(\boldsymbol{v}) = \bigwedge_{\varphi_{\boldsymbol{S}} \in \Phi} \varphi_{\boldsymbol{S}}(\boldsymbol{v}[\boldsymbol{S}])$$

ANITI INRAE

Definition (Markov Random Field (used in Machine Learning, Statistical Physics))

A GM $\mathcal{M} = \langle \boldsymbol{V}, \Phi \rangle$ defined by:

- a sequence of n variables $\boldsymbol{V}$, each with an associated finite domain of size less than d.
- a set Φ of e non negative functions (potentials).

Definition (Joint function and associated probability distribution)

$$\Phi_{\mathcal{M}}(\boldsymbol{v}) = \prod_{\varphi_{\boldsymbol{S}} \in \Phi} \varphi_{\boldsymbol{S}}(\boldsymbol{v}[\boldsymbol{S}]) \qquad P_{\mathcal{M}}(\boldsymbol{V}) \propto \Phi_{\mathcal{M}}(\boldsymbol{V})$$

MRF can be estimated from data

Using eg. regularized approximate/pseudo log-likelihood approaches.

How are functions $\varphi_S \in \Phi$ represented?

- Default: as tensors over B. (multidimensional tables)
- Boolean vars: (weighted) clauses. (disjunction of literals: variables or their negation)
- Using a specific language, subset of all tensors or clauses or dedicated (ALL-DIFFERENT).
- this influences complexities, tensors as a default

ANITI INRAE

A variety of well-studied frameworks

- Propositional Logic (PL): Boolean domains and co-domain, conjunction of clauses
- Constraint Networks (CN): Finite domains, Boolean co-domain, conjunction of tensors
- Cost Function Networks (CFN): Finite domains, numerical co-domain, sum of tensors.
- Markov Random Fields (MRF): Finite domains, $\mathbb{R}^+$ as co-domain, product of tensors.
- Bayesian Networks (BN): MRF + normalized functions and scopes following a DAG.
- Generalized Additive Independence [BG95], Weighted PL, Quadratic Pseudo-Boolean Optimization [BH02]...

ANITI INRAE

Definition ((Hyper)graph of $\mathcal{M} = \langle \boldsymbol{V}, \Phi \rangle$)

One vertex per variable, one (hyper)edge per scope $\boldsymbol{S}$ of function $\varphi_{\boldsymbol{S}} \in \Phi$.

Definition (Factor graph of $\mathcal{M} = \langle \boldsymbol{V}, \Phi \rangle$)

One vertex per variable or function, an edge connects the vertex $\varphi_{\boldsymbol{s}}$ to all variables in $\boldsymbol{S}$.

CFN $\mathcal{M} = \langle \boldsymbol{V}, \Phi \rangle$, parameterized by an upper bound k

$\mathcal{M}$ defines a non negative joint function $\Phi_{\mathcal{M}} = \min(\sum_{\varphi_S \in \Phi} \varphi_S, k)$

Flexible

- $k = 1$ same as Constraint Networks
- $k = \infty$ same as GAI, $-\log()$ transform of MRFs (Boltzmann)
- k finite k is a known upper bound
- $\varphi_{\varnothing}$ is a naive lower bound on the minimum cost

Optimization queries

- SAT/PL: is the minimum of $\Phi_{\mathcal{M}} \preccurlyeq t$?
- CSP/CN: is the minimum of $\Phi_{\mathcal{M}} \preccurlyeq t$?
- WCSP/CFN: is the minimum of $\Phi_{\mathcal{M}} \preccurlyeq \alpha$?
- MAP/MRF: is the minimum of $\Phi_{\mathcal{M}} \preccurlyeq \alpha$?
- MPE/BN: is the minimum of $\Phi_{\mathcal{M}} \preccurlyeq \alpha$?

Counting queries

- #-SAT/PL: how many assignments satisfy $\Phi_{\mathcal{M}} = t$?
- MAR/MRF: compute $Z = \sum(\Phi_{\mathcal{M}})$ or $P_{\mathcal{M}}(X = u)$ where $X \in \boldsymbol{V}$
- MAR/BN: compute $P_{\mathcal{M}}(X = u)$ where $X \in \boldsymbol{V}$

Example: MinCUT with hard and weighted edges

Graph $G = (V, E)$ with edge weight function w

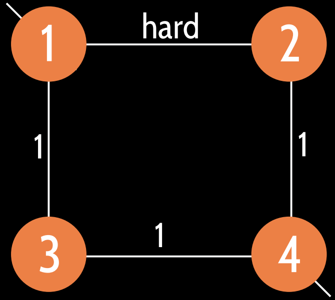
- A boolean variable x_i per vertex $i \in V$
- A cost function $w_{ij} = w(i, j) \times \mathbb{1}[x_i \neq x_j]$ per edge $(i, j) \in E$
- Hard edges: $w_{ij} = k$

EXAMPLE: MINCUT WITH HARD AND WEIGHTED EDGES

Graph $G = (V, E)$ with edge weight function w

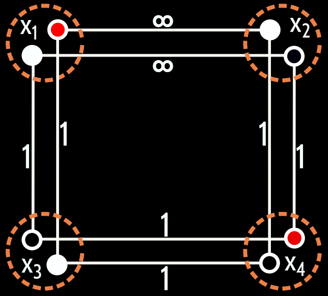
- A boolean variable x_i — per vertex $i \in V$
- A cost function $w_{ij} = w(i, j) \times \mathbb{1}[x_i \neq x_j]$ — per edge $(i, j) \in E$
- Hard edges: $w_{ij} = k$

- vertices $\{1, 2, 3, 4\}$
- cut weights 1
- but edge $(1, 2)$ hard



Graph $G = (V, E)$ with edge weight function w

- A boolean variable x_i — per vertex $i \in V$
- A cost function $w_{ij} = w(i,j) \times \mathbb{1}[x_i \neq x_j]$ — per edge $(i,j) \in E$
- Hard edges: $w_{ij} = k$



- vertices $\{1, 2, 3, 4\}$
- cut weights 1
- but edge $(1, 2)$ hard

ANITI INRAE

MinCut on a 3-clique with hard edge

```
{ problem :{name: MinCut, mustbe: <100.0},
    variables:
      {x1: [l], x2: [l,r], x3: [l,r], x4: [r]}
    functions: {
      cut12:
        {scope: [x1,x2], costs: [0.0, 100.0, 100.0, 0.0]},
      cut13:
        {scope: [x1,x3], costs: [0.0,1.0,1.0,0.0]},
      cut23:
        {scope: [x2,x3], costs: [0.0,1.0,1.0,0.0]}
  ...
}
```

Binary CFN as 01LP (optimisation alone)

The so called "local polytope" [Sch76; Kos99; Wer07] (w/o last line)

$$\text{Function} \sum_{i,a} \varphi_i(a) \cdot x_{ia} + \sum_{\substack{\varphi_{ij} \in \Phi \\ a \in D^i, b \in D^j}} \varphi_{ij}(a,b) \cdot y_{iajb} \quad \text{such that}$$

$$\sum_{a \in D^i} x_{ia} = 1 \qquad \forall i \in \{1, \ldots, n\}$$

$$\sum_{b \in D^j} y_{iajb} = x_{ia} \qquad \forall \varphi_{ij} \in \Phi, \forall a \in D^i$$

$$\sum_{a \in D^i} y_{iajb} = x_{jb} \qquad \forall \varphi_{ij} \in \Phi, \forall b \in D^j$$

$$x_{ia} \in \{0, 1\} \qquad \forall i \in \{1, \ldots, n\}$$

ANITI INRAE

The main algorithmic attractor in the MRF community

- Widely used in image processing (now a bit shadowed by Deep Learning)
- Very large problems: exact approaches considered as unusable [Kap+13].
- Plenty of primal/dual approaches on the local polytope, but universality result [PW13]

Three main families of algorithms

1. global search: backtrack tree-search and branch and bound
2. global inference: non-serial dynamic programming
3. local inference: local application of DP equations

Ignores (useful) stochastic local search approaches.

ANITI INRAE

Time $O(d^n)$, linear space

- If all $|D^X| = 1, \Phi_{\mathcal{M}}(\boldsymbol{v}), \boldsymbol{v} \in D^{\boldsymbol{V}}$ is the answer
- Else choose $X \in \boldsymbol{V}$ s.t. $|D^X| > 1$ and $u \in D^X$ and reduce to
  1. one subproblem where $X_i = u$
  2. one where u is removed from D^X
- Return the minimum of these two subproblems

Branch and Bound

If a lower bound on the optimum is $\succeq$ a known upper bound on $\Phi_{\mathcal{M}}\ldots$ Prune!

NB: $\varphi_{\varnothing}$ is a lower bound, k is our upper bound.

Eliminating variable $X \in \boldsymbol{V}$

Let Φ^X be the set $\{\varphi_{\boldsymbol{S}} \in \Phi \text{ s.t. } X \in \boldsymbol{S}\}$, $\boldsymbol{T}$, the neighbors of X.

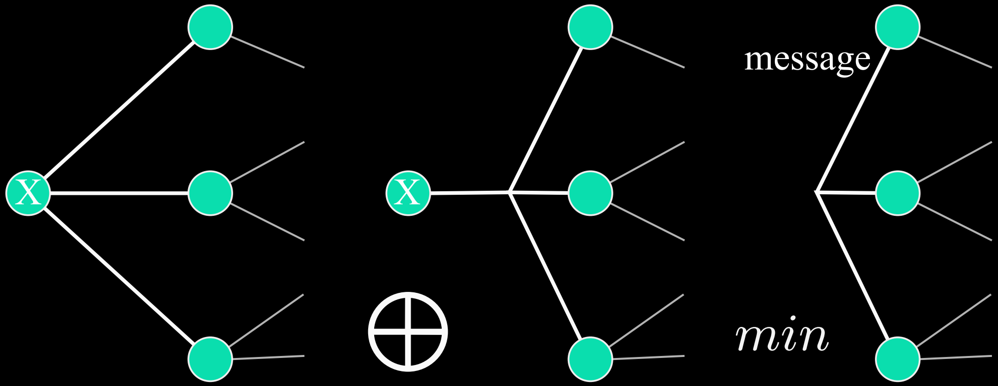
The message $m_{\boldsymbol{T}}^{\Phi_X}$ from Φ^X to $\boldsymbol{T}$ is:

$$m_{\boldsymbol{T}}^{\Phi_X} = \min_X \Big(\bigoplus_{\varphi_{\boldsymbol{S}} \in \Phi^X} \varphi_{\boldsymbol{S}} \Big) \tag{1}$$

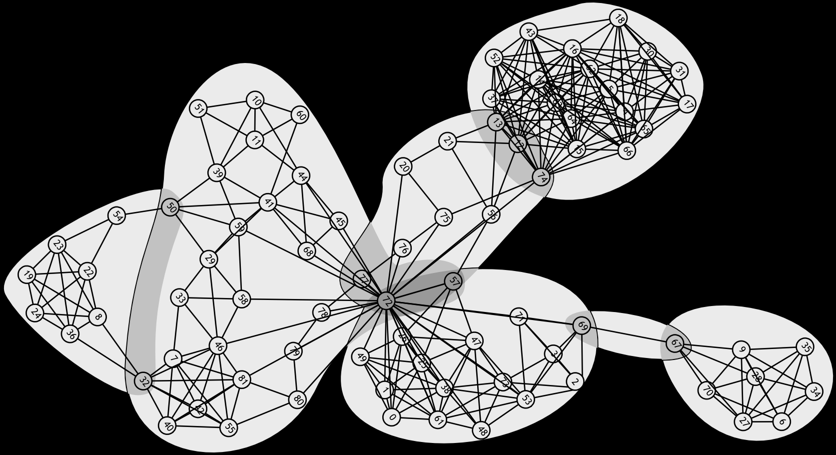
Eliminating a variable — Distributivity

$$\min_{\boldsymbol{v} \in D^{\boldsymbol{V}}} \left[\bigoplus_{\varphi_{\boldsymbol{S}} \in \Phi} (\varphi_{\boldsymbol{S}}(\boldsymbol{v}[\boldsymbol{S}])) \right] = \min_{\boldsymbol{v} \in D^{\boldsymbol{V}-\{X\}}} \left[\bigoplus_{\varphi_{\boldsymbol{S}} \in \Phi - \Phi^X \cup \{m_{\boldsymbol{T}}^{\Phi_X}\}} (\varphi_{\boldsymbol{S}}(\boldsymbol{v}[\boldsymbol{S}])) \right]$$

A graphical representation



A graphical representation



Complexity of one elimination for tensors

Computing $m_{\boldsymbol{T}}^{X}$ is $O(d^{|T+1|})$ time, $O(d^{|T|})$ space $|T|$ is the degree of X

The overall complexity is dominated by the largest degree encountered during elimination

Clauses $\boldsymbol{L}, \boldsymbol{L}'$ clauses

If $\Phi^X = \{(X \vee \boldsymbol{L}), (\neg X \vee \boldsymbol{L'})\}$ $m_{\boldsymbol{T}}^{\Phi_X}$ is $(\boldsymbol{L} \vee \boldsymbol{L'})$.

The resolution principle [Rob65] is an efficient variable elimination process [DR94; DP60].

Exponential in the Dimension [BB69b; BB69a; Bod98] — induced/tree-width

- Dimension of an elimination order for G — Largest set $|T|$ encountered
- Dimension of G — minimum Dimension over all orders
- NP-hard to optimize but useful heuristics exist [BK08].

Tractability

- First tractable class: GMs with bounded tree-width.
- Main approach for exact solving of counting queries for Bayesian nets[LS88].
- Worst case is also best case (space and time)

Message passing

Root the tree and compute messages from leaves

All variables — Variables preserved, time & space $O(ed^2)$

Messages are kept as auxiliary functions.

- When a variable X_i has received messages from all its neighbors but one (X_j)
- Send message m_j^i to X_j

$$m_j^i = \min_{X_i}(\varphi_i \oplus \varphi_{ij} \bigoplus_{X_o \in neigh(X_i), o \neq j} m_i^o) \quad (2)$$

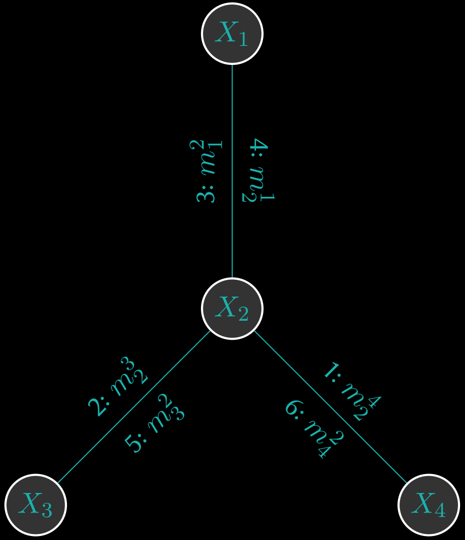


Figure 1: Message passing on a tree, a possible message schedule

The heuristic approach

Starting from e.g., empty messages, apply the message passing equation (2)

$$m_j^i = \min_{X_i}(\varphi_i \oplus \varphi_{ij} \bigoplus_{X_o \in neigh(X_i), o \neq j} m_i^o)$$

on each function until quiescence or maximum number of iterations.

Booleans: Local/arc consistency (CSP), Unit propagation (SAT)

- The unique logically equivalent fixpoint can be efficiently computed
- If it contains $\varphi_\varnothing > 0$, we have a proof of inconsistency

Probabilities: Loopy Belief Propagation [Pea88]

- Often denoted as the "max-sum/min-sum" algorithm.
- At the core of Turbo-decoding [BGT93], implemented in all cell phones.
- Widely studied [YFW01], but known to not always converge.

Equivalence Preserving Transformations

- We can add the message $m^{\Psi}_{\mathbf{Y}}$
- And compensate by 'subtracting' the message from its source

EPTs can enforce generalized versions of "local consistencies"

- Transform the model into an equivalent model
- with a possibly increased $\varphi_{\varnothing}$ (lower bound)
- Reduces to good old *Arc Consistency* in the Boolean case
- Gave birth to Max-resolution in SAT [LH05]

Properties[Coo+10]

- Solves tree-structured problems
- Solves problems with submodular functions (Monge matrices)
- Reduces to a max-flow algorithm on Boolean variables (roof-dual for QPBO)

In the context of local polytope

VAC is a fast incremental approximate solver of the local polytope dual that also enforces AC on logical information

Maintaining LC during Branch and Bound

Combines — Time $O(exp(n))$

- Branch and Bound (Backtrack in the Boolean case)
- Incremental Local Consistency enforcing at each node (lower bound)

Variable (and value) ordering heuristics

- Crucial for empirical efficiency
- Are now adaptive (learned while searching) [Mos+01; Bou+04]
- Little theory.

ANITI INRAE

Additional ingredients

- Search strategies: Best/Depth First [All+15], restarts [GSC97]
- Stronger preprocessing at the root node
- Dominance analysis [Fre91; DPO13; All+14], …
- Conflict directed inference (Boolean) [Bie+09]
- Combined with graph decomposition (tree-decomposition)

SAT solvers

Verification[1], planning, diagnosis, theorem proving,...

[1]Small neural nets too.

[2]Oliver Kullmann. “The Science of Brute Force”. In: *Communications of the ACM* (2017).

SAT solvers

Verification[1], planning, diagnosis, theorem proving,…

2017: proving an "alien" theorem? ∞

When one splits $\mathbb{N}$ in 2, one part must contain a Pythagorean triple $(a^2 = b^2 + c^2)$

[1] Small neural nets too.

[2] Oliver Kullmann. "The Science of Brute Force". In: *Communications of the ACM* (2017).

SAT solvers

Verification[1], planning, diagnosis, theorem proving,…

2017: proving an "alien" theorem? ∞

When one splits $\mathbb{N}$ in 2, one part must contain a Pythagorean triple $(a^2 = b^2 + c^2)$

No known proof, puzzled mathematicians for decades (one offered a 100 $ reward)

[1] Small neural nets too.

[2] Oliver Kullmann. "The Science of Brute Force". In: *Communications of the ACM* (2017).

SAT solvers

Verification[1], planning, diagnosis, theorem proving,…

2017: proving an "alien" theorem? ∞

When one splits $\mathbb{N}$ in 2, one part must contain a Pythagorean triple $(a^2 = b^2 + c^2)$

No known proof, puzzled mathematicians for decades (one offered a 100 $ reward)

SAT solver proof[HKM16; Lam16]

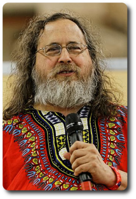
$200TB$ proof, compressed to $86GB$ (stronger proof system)[2]

[1] Small neural nets too.

[2] Oliver Kullmann. "The Science of Brute Force". In: *Communications of the ACM* (2017).

SAT: a lot of free data and free code...

- International competitions ($> 50,000$ benchmarks with many real problems)
- Open source solvers (autocatalytic)



Similar progresses in other "Graphical Model" solvers (CP, CFN)

"ToulBar2 variants were superior to CPLEX variants in all our tests"[HSS18]

(still, there are small problems that cannot be solved in decent time)

VAC vs. LP on Protein design problems

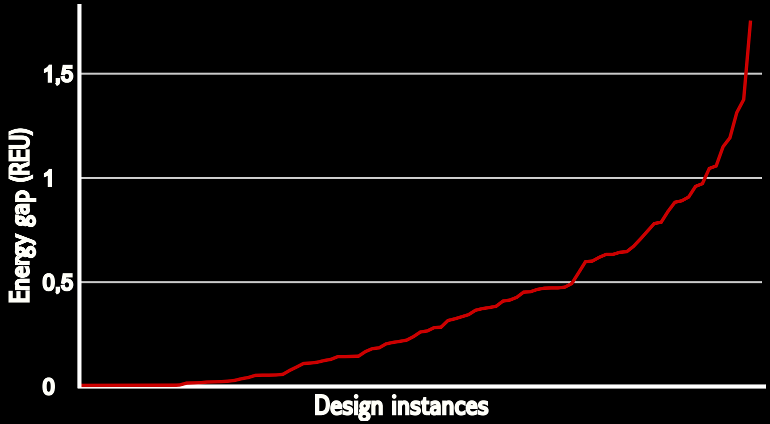
CPLEX V12.4.0.0

```
Problem '3e4h.LP' read.
Root relaxation solution time =  811.28 sec.
...
MIP - Integer optimal solution:  Objective =  150023297067
Solution time =  864.39 sec.
```

tb2 and VAC (AC3 based)

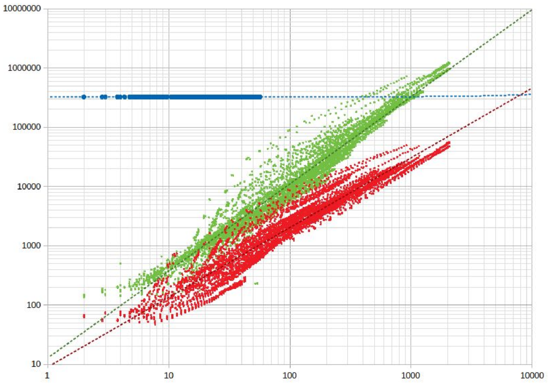
```
loading CFN file: 3e4h.wcsp
Lb after VAC: 150023297067
Preprocessing time: 9.13 seconds.
Optimum: 150023297067 in 129 backtracks, 129 nodes and 9.38 seconds.
```

Comparison with Rosetta's Simulated annealing [Sim+15]



Optimality gap of the Simulated annealing solution as problems get harder

Asymptotic convergence, close to infinity is arbitrarily far



Exact vs. heuristic solvers [Mul+19]

DWave within 1.16 kcal/mol of the optimum 10% of the time, 4.35 kcal/mol 50% of the time, 8.45 kcal/mol 90% of the time.

Learning to play the Sudoku using DL (System 1)

Recent Deep Learning approaches that "learn how to reason"

- Recurrent relational Networks [PPW18]: learn "message passing" like functions
- SAT-net [Wan+19] embeds a convex relaxation of Max2SAT [GW95] as a final differentiable layer

Architecture and prior

- The architectures identify decision variables and (RRN) pairs of interacting variables
- Input: a Sudoku problem (hints)
- Output: a filled Sudoku grid
- Learning: on hint/solution pairs (SGD) (hints: numbers or images, LeNet processed).

Learning to play the Sudoku using GMs (System 2)

ANITI INRAE

Learning MRFs from data

- Optimizing an approximate convex representation of the L1-regularized log-likelihood with ADMM [Par+17]
- Takes expectations of sufficient statistics as input
- Simultaneously estimates the GM graph structure and its parameters (tensors)
- Requires one regularization hyper-parameter λ

In practice

- Adjust λ (empirical risk, using toulbar2) on a test set (1,024 samples)
- Validate (on a separate validation set of 1,000 samples)
- Image hints: use LeNet to transform images to posterior probabilities

Hard and easy problems

- Sodoku instances can be easy (many hints) or hard (17 hints for a unique solution).
- The fraction of solved Sudoku in the validation set depends on their hardness

Different situations

- RRN [PPW18] used 180,000 + 18,000 + 18,000 of problems with varying hardness (17 to 34 hints)
- SATNet [Wan+19]: used 9,000 + 1,000 problems with mostly easy problems (no test set for hyper-parameters tuning)

Results

DL approaches

- RRN: can solve 96.6 % of the hardest Sudokus using 198,000 examples
- SAT-Net can solve 98.3% of easy Sudokus using 10,000 examples

The GM approach learns to solve

- 100 % of hard Sudoku problems from 9,000 + 1,024 examples
- 100 % of easy Sudoku problems from 7,000 + 1,024 examples (58.2% of hard problems)
- The rules of Sudoku can be extracted automatically as constraints [Kum+20]
- These minimum empirically 100% correct GMs do not give “exact” rules
- 13,000 recover an exact formulation of the Sudoku rules

Learning from noisy hints (images)

DL approaches

- RRN: did not try it.
- SAT-Net can solve 63.2 % of easy Sudoku problems from 10,000 samples (theoretical max. of 74.7%: LeNet accuracy 99.2%, 36.2 hints on average)

The GM approach learns to solve

- 82 % of hard Sudoku problems from 8,000+1,024 examples
- 77 % of easy Sudoku problems from 8,000+1.024 examples (more hints, more LeNet errors)
- 13,000 noisy samples are enough to recover an exact formulation of the Sudoku rules

Additional capacities

- one can also use noisy solutions (not only hints) for learning.
- one can add (design) constraints on the output.

Graphical models

- Can be learned from (noisy) data (including DL output if desirable)
- Can often be analyzed and solved using exact (or guaranteed) algorithms
- theoreticals limits[Vuf+16], PAC learnability [Kum+20], specialized languages?

Thank You!

Questions?

David Allouche et al. “Computational protein design as an optimization problem”. In: *Artificial Intelligence* 212 (2014), pp. 59–79.

David Allouche et al. “Anytime Hybrid Best-First Search with Tree Decomposition for Weighted CSP”. In: *Principles and Practice of Constraint Programming*. Springer. 2015, pp. 12–29.

Srinivas M Aji and Robert J McEliece. “The generalized distributive law”. In: *IEEE transactions on Information Theory* 46.2 (2000), pp. 325–343.

Umberto Bertele and Francesco Brioschi. “A new algorithm for the solution of the secondary optimization problem in non-serial dynamic programming”. In: *Journal of Mathematical Analysis and Applications* 27.3 (1969), pp. 565–574.

Umberto Bertele and Francesco Brioschi. “Contribution to nonserial dynamic programming”. In: *Journal of Mathematical Analysis and Applications* 28.2 (1969), pp. 313–325.

Umberto Bertelé and Francesco Brioshi. *Nonserial Dynamic Programming*. Academic Press, 1972.

Fahiem Bacchus and Adam Grove. “Graphical models for preference and utility”. In: *Proceedings of the Eleventh conference on Uncertainty in artificial intelligence.* Morgan Kaufmann Publishers Inc. 1995, pp. 3–10.

Claude Berrou, Alain Glavieux, and Punya Thitimajshima. “Near Shannon limit error-correcting coding and decoding: Turbo-codes. 1”. In: *Proceedings of ICC’93-IEEE International Conference on Communications.* Vol. 2. IEEE. 1993, pp. 1064–1070.

E. Boros and P. Hammer. “Pseudo-Boolean Optimization”. In: *Discrete Appl. Math.* 123 (2002), pp. 155–225.

Armin Biere et al. “Conflict-driven clause learning sat solvers”. In: *Handbook of Satisfiability, Frontiers in Artificial Intelligence and Applications* (2009), pp. 131–153.

H L Bodlaender and A M C A Koster. *Treewidth Computations I. Upper Bounds.* Tech. rep. UU-CS-2008-032. Utrecht, The Netherlands: Utrecht University, Department of Information and Computing Sciences, Sept. 2008. URL: `http://www.cs.uu.nl/research/techreps/repo/CS-2008/2008-032.pdf`.

Hans L Bodlaender. “A partial k-arboretum of graphs with bounded treewidth”. In: *Theoretical computer science* 209.1-2 (1998), pp. 1–45.

Frédéric Boussemart et al. “Boosting systematic search by weighting constraints”. In: *ECAI*. Vol. 16. 2004, p. 146.

M. Cooper et al. “Soft arc consistency revisited”. In: *Artificial Intelligence* 174 (2010), pp. 449–478.

Rina Dechter. “Bucket Elimination: A Unifying Framework for Reasoning”. In: *Artificial Intelligence* 113.1–2 (1999), pp. 41–85.

Martin Davis and Hilary Putnam. “A computing procedure for quantification theory”. In: *Journal of the ACM (JACM)* 7.3 (1960), pp. 201–215.

Simon De Givry, Steven D Prestwich, and Barry O’Sullivan. “Dead-end elimination for weighted CSP”. In: *Principles and Practice of Constraint Programming*. Springer. 2013, pp. 263–272.

Rina Dechter and Irina Rish. “Directional resolution: The Davis-Putnam procedure, revisited”. In: *KR* 94 (1994), pp. 134–145.

Eugene C. Freuder. “Eliminating Interchangeable Values in Constraint Satisfaction Problems”. In: *Proc. of AAAI’91*. Anaheim, CA, 1991, pp. 227–233.

Carla P Gomes, Bart Selman, and Nuno Crato. “Heavy-tailed distributions in combinatorial search”. In: *International Conference on Principles and Practice of Constraint Programming*. Springer. 1997, pp. 121–135.

Michel X Goemans and David P Williamson. “Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming”. In: *Journal of the ACM (JACM)* 42.6 (1995), pp. 1115–1145.

Marijn JH Heule, Oliver Kullmann, and Victor W Marek. “Solving and verifying the boolean pythagorean triples problem via cube-and-conquer”. In: *International Conference on Theory and Applications of Satisfiability Testing*. Springer. 2016, pp. 228–245.

Stefan Haller, Paul Swoboda, and Bogdan Savchynskyy. “Exact MAP-Inference by Confining Combinatorial Search with LP Relaxation”. In: *Thirty-Second AAAI Conference on Artificial Intelligence*. 2018.

Joerg Kappes et al. “A comparative study of modern inference techniques for discrete energy minimization problems”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2013, pp. 1328–1335.

Vladimir Kolmogorov. “Convergent tree-reweighted message passing for energy minimization”. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 28.10 (2006), pp. 1568–1583.

A M C A. Koster. “Frequency assignment: Models and Algorithms”. Available at www.zib.de/koster/thesis.html. PhD thesis. The Netherlands: University of Maastricht, Nov. 1999.

Oliver Kullmann. “The Science of Brute Force”. In: *Communications of the ACM* (2017).

Mohit Kumar et al. “Learning MAX-SAT from Contextual Examples for Combinatorial Optimisation”. In: *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence*. AAAI. 2020.

Evelyn Lamb. "Maths proof smashes size record: supercomputer produces a 200-terabyte proof–but is it really mathematics?" In: *Nature* 534.7605 (2016), pp. 17–19.

J. Larrosa and F. Heras. "Resolution in Max-SAT and its relation to local consistency in weighted CSPs". In: *Proc. of the 19th IJCAI*. Edinburgh, Scotland, 2005, pp. 193–198.

S.L. Lauritzen and D.J. Spiegelhalter. "Local computations with probabilities on graphical structures and their application to expert systems". In: *Journal of the Royal Statistical Society – Series B* 50 (1988), pp. 157–224.

Matthew W Moskewicz et al. "Chaff: Engineering an efficient SAT solver". In: *Proceedings of the 38th annual Design Automation Conference*. ACM. 2001, pp. 530–535.

Vikram Khipple Mulligan et al. "Designing Peptides on a Quantum Computer". In: *bioRxiv* (2019), p. 752485.

Youngsuk Park et al. “Learning the network structure of heterogeneous data via pairwise exponential Markov random fields”. In: *Proceedings of machine learning research* 54 (2017), p. 1302.

Judea Pearl. *Probabilistic Reasoning in Intelligent Systems, Networks of Plausible Inference.* Palo Alto: Morgan Kaufmann, 1988.

Rasmus Palm, Ulrich Paquet, and Ole Winther. “Recurrent relational networks”. In: *Advances in Neural Information Processing Systems.* 2018, pp. 3368–3378.

Daniel Prusa and Tomas Werner. “Universality of the local marginal polytope”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 2013, pp. 1738–1743.

J. Alan Robinson. “A machine-oriented logic based on the resolution principle”. In: *Journal of the ACM* 12 (1965), pp. 23–44.

T. Schiex. “Arc consistency for soft constraints”. In: *Principles and Practice of Constraint Programming - CP 2000.* Vol. 1894. LNCS. Singapore, Sept. 2000, pp. 411–424.

M.I. Schlesinger. “Sintaksicheskiy analiz dvumernykh zritelnikh signalov v usloviyakh pomekh (Syntactic analysis of two-dimensional visual signals in noisy conditions)”. In: *Kibernetika* 4 (1976), pp. 113–130.

G. Shafer. *An Axiomatic Study of Computation in Hypertrees.* Working paper 232. Lawrence: University of Kansas, School of Business, 1991.

David Simoncini et al. “Guaranteed Discrete Energy Optimization on Large Protein Design Problems”. In: *Journal of Chemical Theory and Computation* 11.12 (2015), pp. 5980–5989. DOI: `10.1021/acs.jctc.5b00594`.

Marc Vuffray et al. “Interaction screening: Efficient and sample-optimal learning of Ising models”. In: *Advances in Neural Information Processing Systems.* 2016, pp. 2595–2603.

Po-Wei Wang et al. “SATNet: Bridging deep learning and logical reasoning using a differentiable satisfiability solver”. In: *ICML’19 proceedings, arXiv preprint arXiv:1905.12149.* 2019.

T. Werner. "A Linear Programming Approach to Max-sum Problem: A Review.". In: *IEEE Trans. on Pattern Recognition and Machine Intelligence* 29.7 (July 2007), pp. 1165–1179. URL: http://dx.doi.org/10.1109/TPAMI.2007.1036.

Jonathan S Yedidia, William T Freeman, and Yair Weiss. "Bethe free energy, Kikuchi approximations, and belief propagation algorithms". In: *Advances in neural information processing systems* 13 (2001).