Constrained Optimization over Semirings

Kuldeep S. Meel

University of Toronto

Joint work with Arnab Bhattacharyya, Pavan Aduri, and N. V. Vinodchandran

Relevant publication: AAAI-23

Acknowledgements: Val Tannen for introducing us to this problem and the world of semirings at large

Boolean Interpretations

$F := (x_1) \wedge (x_2) \wedge (\neg x_1 \vee \neg x_2)$

SAT: Is there is a truth assignment to the variables so that F is evaluated to True.

Boolean Interpretations

$F := (x_1) \wedge (x_2) \wedge (\neg x_1 \vee \neg x_2)$

SAT: Is there is a truth assignment to the variables so that F is evaluated to True.

- Boolean Interpretation
  - $K = \{0, 1\}$
  - $\neg :=$ NOT function
  - $\wedge :=$ AND function
  - $\vee :=$ OR function

SAT: Compute $\max_\pi\{\pi(F)\}$ over all interpretations $\pi : X \rightarrow K$.

X is the set of variables and $\pi(F)$ is the natural extension of π to F.

F is satisfiable if and only if $\max_\pi\{\pi(F)\} = 1$.

Beyond Boolean Interpretations

$F := (x_1) \wedge (x_2) \wedge (\neg x_1 \vee \neg x_2)$

Viterbi semiring interpretation

- $K = [0, 1]$
- $\wedge :=$ MULT function
- $\vee :=$ MAX function
- $\neg x := 1 - x$

Problem: Given F: Compute $\max_{\pi}\{\pi(F)\}$ over all interpretations $\pi : X \to K$.

Beyond Boolean Interpretations

$F := (x_1) \wedge (x_2) \wedge (\neg x_1 \vee \neg x_2)$

Viterbi semiring interpretation

- $K = [0, 1]$
- $\wedge :=$ MULT function
- $\vee :=$ MAX function
- $\neg x := 1 - x$

Problem: Given F: Compute $\max_\pi\{\pi(F)\}$ over all interpretations $\pi : X \to K$.

For F above:

- $\max\{x_1 x_2(1 - x_1), x_1 x_2(1 - x_2)\}$
- $\pi(x_1) = 0.5$; $\pi(x_2) = 1$
- $\pi(F) = 0.5 \cdot 1 \cdot 0.5 = \boxed{0.25}$

Beyond Boolean Interpretations

$F := (x_1) \wedge (x_2) \wedge (\neg x_1 \vee \neg x_2)$

Viterbi semiring interpretation

- $K = [0, 1]$
- $\wedge :=$ MULT function
- $\vee :=$ MAX function
- $\neg x := 1 - x$

Problem: Given F: Compute $\max_\pi\{\pi(F)\}$ over all interpretations $\pi : X \to K$.

For F above:

- $\max\{x_1 x_2(1 - x_1), x_1 x_2(1 - x_2)\}$
- $\pi(x_1) = 0.5$; $\pi(x_2) = 1$
- $\pi(F) = 0.5 \cdot 1 \cdot 0.5 = \boxed{0.25}$

F is satisfiable (Boolean) $\Leftrightarrow \max_\pi\{\pi(F)\} = 1$ (Viterbi)

Interpretation of Negation

How do we interpret $\neg : K \to K$?

$\neg(x) = 1 - x$ is one of them.

For our upper bounds any "reasonable" interpretation of negation suffice.

$$\begin{aligned} \neg\neg(x) &= x \\ \neg(0) &= 1 \end{aligned}$$

Useful Semirings

- Viterbi semiring $\mathbb{V} = ([0,1], \max, \cdot, 0, 1)$.
  - Database provenance, where $x \in [0,1]$ is interpreted as a *confidence score*.
  - Probabilistic parsing, probabilistic CSPs, Hidden Markov Models.

- Tropical semiring $\mathbb{T} = (\mathbb{R} \cup \{\infty\}, \min, +, \infty, 0)$.
  - Cost analysis and algebraic formulation for shortest path algorithms.

- Fuzzy semiring $\mathbb{F} = ([0,1], \max, \min, 0, 1)$.

- Access control semiring $\mathbb{A}_k = ([k], \max, \min, 0, k)$
  - Security Specification. Each $i \in [k]$ is associated with a access control level with natural ordering. 0 corresponds to public access and k corresponds to no access at all.

Computational Problem: OptVal

For a given semiring K and input formula F (in negation normal form)

OptVal: Compute $\max_{\pi}\{\pi(F)\}$ over all interpretations $\pi : X \to K$.

What is the complexity of OptVal?

Computational Problem: OptVal

For a given semiring K and input formula F (in negation normal form)

OptVal: Compute $\max_\pi\{\pi(F)\}$ over all interpretations $\pi : X \to K$.

What is the complexity of OptVal?

- Long history of work focused on development of practical tools in CSP community
- (Surprisingly) No prior work from computational complexity perspective for cases other than Boolean semiring

Our Results (AAAI-23)

Fuzzy, Access Control Same as Boolean case

Viterbi, Tropical $\mathsf{FP}^{\mathsf{NP}[\log]} \leq \mathsf{OptVal} \leq \mathsf{FP}^{\mathsf{NP}}$.

And the proof arguments are really simple and beautiful (I am, of course, biased!)

Upperbound: OptVal $\in$ FP$^{\text{NP}}$

Define a binary search language $L_{opt} = \{\langle F, v \rangle \mid \text{OptVal}(F) \geq v\}$.

- Perform binary search over $[0, 1]$ by making queries to L_{opt}

Upperbound: OptVal $\in \mathsf{FP}^{\mathsf{NP}}$

Define a binary search language $L_{opt} = \{\langle F, v\rangle \mid \mathrm{OptVal}(F) \geq v\}$.

- Perform binary search over $[0, 1]$ by making queries to L_{opt}

  Challenge: OptVal(F) could potentially be any real number. Do not know when to stop the binary search.

Upperbound: OptVal $\in \mathsf{FP}^{\mathsf{NP}}$

Define a binary search language $L_{opt} = \{\langle F, v\rangle \mid \mathrm{OptVal}(F) \geq v\}$.

- Perform binary search over $[0, 1]$ by making queries to L_{opt}

  Challenge: OptVal(F) could potentially be any real number. Do not know when to stop the binary search.

  Example: $F := (x_1) \wedge (x_2) \wedge (\neg x_1 \vee \neg x_2)$

- Consider the optimal interpretation $\hat{\pi}$ and suppose we know which literal takes the maximum value in each of the clauses under $\hat{\pi}$.

Upperbound: OptVal $\in \mathsf{FP}^{\mathsf{NP}}$

Define a binary search language $L_{opt} = \{\langle F, v \rangle \mid \mathrm{OptVal}(F) \geq v\}$.

- Perform binary search over $[0, 1]$ by making queries to L_{opt}

  Challenge: OptVal(F) could potentially be any real number. Do not know when to stop the binary search.

  Example: $F := (x_1) \wedge (x_2) \wedge (\neg x_1 \vee \neg x_2)$

- Consider the optimal interpretation $\hat{\pi}$ and suppose we know which literal takes the maximum value in each of the clauses under $\hat{\pi}$.
  - Say $\hat{\pi}(\neg x_1 \vee \neg x_2)) = \hat{\pi}(\neg x_2)$, i.e., $\neg x_2$ takes the maximum value in the clause $(\neg x_1 \vee \neg x_2)$.
  - $\hat{\pi}(F) = \hat{\pi}(x_1) \cdot \hat{\pi}(x_2) \cdot \hat{\pi}(\neg x_2) = \hat{\pi}(x_1) \cdot \hat{\pi}(x_2) \cdot (1 - \hat{\pi}(x_2))$

Upperbound: OptVal $\in$ FP$^{\text{NP}}$

Define a binary search language $L_{opt} = \{\langle F, v\rangle \mid \text{OptVal}(F) \geq v\}$.

- Perform binary search over $[0, 1]$ by making queries to L_{opt}

  Challenge: OptVal(F) could potentially be any real number. Do not know when to stop the binary search.

  Example: $F := (x_1) \wedge (x_2) \wedge (\neg x_1 \vee \neg x_2)$

- Consider the optimal interpretation $\hat{\pi}$ and suppose we know which literal takes the maximum value in each of the clauses under $\hat{\pi}$.
  - Say $\hat{\pi}(\neg x_1 \vee \neg x_2)) = \hat{\pi}(\neg x_2)$, i.e., $\neg x_2$ takes the maximum value in the clause $(\neg x_1 \vee \neg x_2)$.
  - $\hat{\pi}(F) = \hat{\pi}(x_1) \cdot \hat{\pi}(x_2) \cdot \hat{\pi}(\neg x_2) = \hat{\pi}(x_1) \cdot \hat{\pi}(x_2) \cdot (1 - \hat{\pi}(x_2))$
  - $\hat{\pi}(x_1) = 1$ and $\hat{\pi}(x_2) = 0.5$
- Let x_i and $\neg x_i$ takes maximum value in ℓ_i and k_i clauses respectively

- Observation: $\hat{\pi}(F) = \prod_{x_i} \hat{\pi}(x_i)^{\ell_i}(1 - \hat{\pi}(x_i))^{k_i}$

Upperbound: OptVal $\in$ FP$^{\text{NP}}$

Define a binary search language $L_{opt} = \{\langle F, v\rangle \mid \text{OptVal}(F) \geq v\}$.

- Perform binary search over $[0, 1]$ by making queries to L_{opt}

  Challenge: OptVal(F) could potentially be any real number. Do not know when to stop the binary search.

  Example: $F := (x_1) \wedge (x_2) \wedge (\neg x_1 \vee \neg x_2)$

- Consider the optimal interpretation $\hat{\pi}$ and suppose we know which literal takes the maximum value in each of the clauses under $\hat{\pi}$.
  - Say $\hat{\pi}(\neg x_1 \vee \neg x_2)) = \hat{\pi}(\neg x_2)$, i.e., $\neg x_2$ takes the maximum value in the clause $(\neg x_1 \vee \neg x_2)$.
  - $\hat{\pi}(F) = \hat{\pi}(x_1) \cdot \hat{\pi}(x_2) \cdot \hat{\pi}(\neg x_2) = \hat{\pi}(x_1) \cdot \hat{\pi}(x_2) \cdot (1 - \hat{\pi}(x_2))$
  - $\hat{\pi}(x_1) = 1$ and $\hat{\pi}(x_2) = 0.5$
- Let x_i and $\neg x_i$ takes maximum value in ℓ_i and k_i clauses respectively

- Observation: $\hat{\pi}(F) = \prod_{x_i} \hat{\pi}(x_i)^{\ell_i}(1 - \hat{\pi}(x_i))^{k_i} = \prod_{x_i} \left(\frac{\ell_i}{\ell_i + k_i}\right)^{\ell_i} \cdot \left(\frac{k_i}{\ell_i + k_i}\right)^{k_i}$

Upperbound: OptVal $\in \mathsf{FP}^{\mathsf{NP}}$

Define a binary search language $L_{opt} = \{\langle F, v\rangle \mid \text{OptVal}(F) \geq v\}$.

- Perform binary search over $[0, 1]$ by making queries to L_{opt}

  Challenge: OptVal(F) could potentially be any real number. Do not know when to stop the binary search.

  Example: $F := (x_1) \wedge (x_2) \wedge (\neg x_1 \vee \neg x_2)$

- Consider the optimal interpretation $\hat{\pi}$ and suppose we know which literal takes the maximum value in each of the clauses under $\hat{\pi}$.
  - Say $\hat{\pi}(\neg x_1 \vee \neg x_2)) = \hat{\pi}(\neg x_2)$, i.e., $\neg x_2$ takes the maximum value in the clause $(\neg x_1 \vee \neg x_2)$.
  - $\hat{\pi}(F) = \hat{\pi}(x_1) \cdot \hat{\pi}(x_2) \cdot \hat{\pi}(\neg x_2) = \hat{\pi}(x_1) \cdot \hat{\pi}(x_2) \cdot (1 - \hat{\pi}(x_2))$
  - $\hat{\pi}(x_1) = 1$ and $\hat{\pi}(x_2) = 0.5$
- Let x_i and $\neg x_i$ takes maximum value in ℓ_i and k_i clauses respectively

- Observation: $\hat{\pi}(F) = \prod_{x_i} \hat{\pi}(x_i)^{\ell_i}(1 - \hat{\pi}(x_i))^{k_i} = \prod_{x_i} \left(\frac{\ell_i}{\ell_i+k_i}\right)^{\ell_i} \cdot \left(\frac{k_i}{\ell_i+k_i}\right)^{k_i}$

- Lemma: OptVal$(F) \in \mathcal{C}_N$ for $N \in 2^{\mathrm{poly}(size(F))}$.

  $\mathcal{C}_N$: Farey Sequence of order N. Fractions of the form A/B, where $1 \leq A, B \leq N$ and $\gcd(A, B) = 1$.

Hardness for Viterbi: MaxSAT $\leq$ OptVal

Confidence Bounding Lemma: Let F be a CNF formula with m clauses and r the maximum number of satisfiable clauses (over the Boolean semiring). Then,

$$\hat{\pi}(F) \leq \frac{1}{4^{m-r}}$$

Hardness for Viterbi: MaxSAT $\leq$ OptVal

Confidence Bounding Lemma: Let F be a CNF formula with m clauses and r the maximum number of satisfiable clauses (over the Boolean semiring). Then,

$$\hat{\pi}(F) \leq \frac{1}{4^{m-r}}$$

Hardness for Viterbi: MaxSAT $\leq$ OptVal

Confidence Bounding Lemma: Let F be a CNF formula with m clauses and r the maximum number of satisfiable clauses (over the Boolean semiring). Then,

$$\hat{\pi}(F) \leq \frac{1}{4^{m-r}}$$

Reduction $F \to F'$: $C_i \to (C_i \vee y_i) \wedge (\neg y_i)$ for each i

Hardness for Viterbi: MaxSAT $\leq$ OptVal

Confidence Bounding Lemma: Let F be a CNF formula with m clauses and r the maximum number of satisfiable clauses (over the Boolean semiring). Then,

$$\hat{\pi}(F) \leq \frac{1}{4^{m-r}}$$

Reduction $F \to F'$: $C_i \to (C_i \vee y_i) \wedge (\neg y_i)$ for each i

Claim: $\mathrm{OptVal}(F') = 1/4^{m-r}$

- We can give an interpretation π so that $\pi(F') = 1/4^{m-r}$.
- That is the best possible since
  - number of clauses of $F' = 2m$
  - maximum number of clauses that can be satisfied is $m + r$

Where we are and where do we go from here? – I

MaxSAT $\leq$ OptVal

Speculative Thoughts

- OptVal can be expressed as sum of logs of max over real-valued variables?
- Can this be a natural problem that's more suited for continuous methods such as Neural Networks?
- So a possibility would be to start with a MaxSAT problem, generate the corresponding OptVal problem and use a continuous method to solve it and then recover the answer.

Where we are and where do we go from here? – II

$$\mathsf{FP}^{\mathsf{NP[log]}} \leq \mathsf{OptVal} \leq \mathsf{FP}^{\mathsf{NP}}$$

Can we close the gap?

Two possibilities

- OptVal $\in \mathsf{FP}^{\mathsf{NP[log]}}$
  - Rely on the progress in MaxSAT solving to build practical tools
  - Open up questions regarding optimal encoding to MaxSAT and if specialized algorithms can outperform MaxSAT-based approaches

Where we are and where do we go from here? – II

$$\mathsf{FP}^{\mathsf{NP[log]}} \leq \mathsf{OptVal} \leq \mathsf{FP}^{\mathsf{NP}}$$

Can we close the gap?

Two possibilities

- OptVal $\in$ $\mathsf{FP}^{\mathsf{NP[log]}}$
  - Rely on the progress in MaxSAT solving to build practical tools
  - Open up questions regarding optimal encoding to MaxSAT and if specialized algorithms can outperform MaxSAT-based approaches
- OptVal is $\mathsf{FP}^{\mathsf{NP}}$-hard
  - Well, a natural problem that's complete for $\mathsf{FP}^{\mathsf{NP}}$
  - How do we design practical algorithms that can rely on the progress in SAT solving?
  - Binary search-based techniques didn't work well for MaxSAT.

In summary: The future is exciting either way!

These slides are available at `www.cs.toronto.edu/~meel/talks.html`

Backup

L_{opt} is in NP

- Represent the NNF formula as a formula tree F
- Proof Tree of a formula: For every OR node in F keep one of the subtrees. For every AND node keep both.
- optSemVal of a proof tree is of the form $\left(\frac{a}{a+b}\right)^a \cdot \left(\frac{b}{a+b}\right)^b$.
- optSemVal(ϕ) is the maximum over optSemVal(T) over all proof trees T.
- NP Algorithm: Guess a proof tree T and compute its optSemVal.

Algorithm

- Perform Binary search using L_{opt} till we find an interval $[L, R]$ with $R - L \leq 1/N$.
- Find a member of $\mathcal{F}_N$ that lies in the interval $[L, R]$.
- Use NP calls to an appropriately defined NP language over Farey sequences.