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

\[ F := (x_1) \land (x_2) \land (\neg x_1 \lor \neg x_2) \]

**SAT**: Is there is a truth assignment to the variables so that \( F \) is evaluated to True.
Boolean Interpretations

\[ F := (x_1) \wedge (x_2) \wedge (\neg x_1 \lor \neg x_2) \]

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

- **Boolean Interpretation**
  - \( K = \{0, 1\} \)
  - \( \neg \) := NOT function
  - \( \wedge \) := AND function
  - \( \lor \) := 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 \( \pi \) to \( F \).

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

\[ F := (x_1) \land (x_2) \land (\neg x_1 \lor \neg x_2) \]

Viterbi semiring interpretation

- \( K = [0, 1] \)
- \( \land := \text{MULT function} \)
- \( \lor := \text{MAX function} \)
- \( \neg x := 1 - x \)

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

\[ F := (x_1) \land (x_2) \land (\neg x_1 \lor \neg x_2) \]

Viterbi semiring interpretation

- \( K = [0, 1] \)
- \( \land := \text{MULT function} \)
- \( \lor := \text{MAX function} \)
- \( \neg x := 1 - x \)

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

For \( F \) above:

- \( \max \{x_1 x_2(1 - x_1), x_1 x_2(1 - x_2)\} \)
- \( \pi(x_1) = 0.5; \ \pi(x_2) = 1 \)
- \( \pi(F) = 0.5 \cdot 1 \cdot 0.5 = 0.25 \)
Beyond Boolean Interpretations

\[ F := (x_1) \land (x_2) \land (\neg x_1 \lor \neg x_2) \]

Viterbi semiring interpretation

- \( K = [0, 1] \)
- \( \land := \text{MULT function} \)
- \( \lor := \text{MAX function} \)
- \( \neg x := 1 - x \)

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

For \( F \) above:

- \( \max \{ x_1x_2(1 - x_1), x_1x_2(1 - x_2) \} \)
- \( \pi(x_1) = 0.5; \pi(x_2) = 1 \)
- \( \pi(F) = 0.5 \cdot 1 \cdot 0.5 = \boxed{0.25} \)

\( F \) is satisfiable (Boolean) \( \Leftrightarrow \max \pi \{ \pi(F) \} = 1 \) (Viterbi)
How do we interpret $\neg : K \rightarrow K$?

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

For our upper bounds any “reasonable” interpretation of negation suffice.

\[ \neg \neg(x) = x \]
\[ \neg(0) = 1 \]
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)

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

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** $\text{FP}^{\text{NP}[\log]} \leq \text{OptVal} \leq \text{FP}^{\text{NP}}$.

And the proof arguments are really simple and beautiful (I am, of course, biased!)
Upperbound: \( \text{OptVal} \in \text{FP}^{\text{NP}} \)

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 \( L_{opt} \)

\[ \text{Challenge: OptVal}(F) \text{ could potentially be any real number. Do not know when to stop the binary search.} \]

\[ \text{Example: } F := (x_1) \land (x_2) \land (\neg x_1 \lor \neg x_2) \]

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

  - \( \hat{\pi}(\neg x_1 \lor \neg x_2)) = \hat{\pi}(\neg x_2) \), i.e., \( \neg x_2 \) takes the maximum value in the clause \( (\neg x_1 \lor \neg x_2) \).

  - \( \hat{\pi}(F) = \hat{\pi}(x_1) \cdot \hat{\pi}(x_2) \cdot (1 - \hat{\pi}(x_2)) = \hat{\pi}(x_1) \cdot \hat{\pi}(x_2) \cdot (1 - \hat{\pi}(x_2)) \).

  - \( \hat{\pi}(x_1) = 1 \) and \( \hat{\pi}(x_2) = 0 \).

\[ \text{Observation: } \hat{\pi}(F) = Q^i x_i \hat{\pi}(x_i) \ell_i (1 - \hat{\pi}(x_i)) k_i \ell_i + k_i = Q^i x_i \ell_i \ell_i + k_i k_i \ell_i \cdot k_i \ell_i \]

\[ \text{Lemma: OptVal}(F) \in C_{N} \text{ for } N \in 2^{\text{poly}(\text{size}(F))} \).

\( C_{N} \): Farey Sequence of order \( N \). Fractions of the form \( A/B \), where \( 1 \leq A, B \leq N \) and \( \gcd(A, B) = 1 \).
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.
Upperbound: $\text{OptVal} \in \text{FP}^{\text{NP}}$

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

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

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

  **Example:** $F := (x_1) \land (x_2) \land (\neg x_1 \lor \neg x_2)$

- Consider the optimal interpretation $\hat{\pi}$ and suppose we know which literal takes the maximum value in each of the clauses under $\hat{\pi}$. 
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:** $F := (x_1) \land (x_2) \land (\neg x_1 \lor \neg x_2)$
- 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 $\hat{\pi}(\neg x_1 \lor \neg x_2) = \hat{\pi}(\neg x_2)$, i.e., $\neg x_2$ takes the maximum value in the clause $(\neg x_1 \lor \neg x_2)$.
  - $\hat{\pi}(F) = \hat{\pi}(x_1) \cdot \hat{\pi}(x_2) \cdot \hat{\pi}(\neg x_2) = \hat{\pi}(x_1) \cdot \hat{\pi}(x_2) \cdot (1 - \hat{\pi}(x_2))$
Upperbound: \( \text{OptVal} \in \mathsf{FP}^{\mathsf{NP}} \)

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

- Perform binary search over \([0, 1]\) by making queries to \( L_{\text{opt}} \)

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

**Example:** \( F := (x_1) \land (x_2) \land (\neg x_1 \lor \neg x_2) \)

- 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 \( \hat{\pi}(\neg x_1 \lor \neg x_2) = \hat{\pi}(\neg x_2) \), i.e., \( \neg x_2 \) takes the maximum value in the clause \((\neg x_1 \lor \neg x_2)\).
  - \( \hat{\pi}(F) = \hat{\pi}(x_1) \cdot \hat{\pi}(x_2) \cdot \hat{\pi}(\neg x_2) = \hat{\pi}(x_1) \cdot \hat{\pi}(x_2) \cdot (1 - \hat{\pi}(x_2)) \)
  - \( \hat{\pi}(x_1) = 1 \) and \( \hat{\pi}(x_2) = 0.5 \)

- Let \( x_i \) and \( \neg x_i \) takes maximum value in \( \ell_i \) and \( k_i \) clauses respectively

- **Observation:** \( \hat{\pi}(F) = \prod_{x_i} \hat{\pi}(x_i)^{\ell_i} (1 - \hat{\pi}(x_i))^{k_i} \)
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:** \( \text{OptVal}(F) \) could potentially be any real number. Do not know when to stop the binary search.

**Example:** \( F := (x_1) \wedge (x_2) \wedge (\neg x_1 \lor \neg x_2) \)

- 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 \( \hat{\pi}(\neg x_1 \lor \neg x_2)) = \hat{\pi}(\neg x_2) \), i.e., \( \neg x_2 \) takes the maximum value in the clause \( (\neg x_1 \lor \neg x_2) \).
  
  - \( \hat{\pi}(F) = \hat{\pi}(x_1) \cdot \hat{\pi}(x_2) \cdot \hat{\pi}(\neg x_2) = \hat{\pi}(x_1) \cdot \hat{\pi}(x_2) \cdot (1 - \hat{\pi}(x_2)) \)
  
  - \( \hat{\pi}(x_1) = 1 \) and \( \hat{\pi}(x_2) = 0.5 \)

- Let \( x_i \) and \( \neg x_i \) takes maximum value in \( \ell_i \) and \( k_i \) clauses respectively

- **Observation:**
  
  \[
  \hat{\pi}(F) = \prod_{x_i} \hat{\pi}(x_i)^{\ell_i} (1 - \hat{\pi}(x_i))^{k_i} = \prod_{x_i} \left( \frac{\ell_i}{\ell_i + k_i} \right)^{\ell_i} \cdot \left( \frac{k_i}{\ell_i + k_i} \right)^{k_i}
  \]
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: \( \text{OptVal}(F) \) could potentially be any real number. Do not know when to stop the binary search.

Example: \( F := (x_1) \land (x_2) \land (\neg x_1 \lor \neg x_2) \)

- 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 \( \hat{\pi}(\neg x_1 \lor \neg x_2)) = \hat{\pi}(\neg x_2) \), i.e., \( \neg x_2 \) takes the maximum value in the clause \( (\neg x_1 \lor \neg x_2) \).
  - \( \hat{\pi}(F) = \hat{\pi}(x_1) \cdot \hat{\pi}(x_2) \cdot \hat{\pi}(\neg x_2) = \hat{\pi}(x_1) \cdot \hat{\pi}(x_2) \cdot (1 - \hat{\pi}(x_2)) \)
  - \( \hat{\pi}(x_1) = 1 \) and \( \hat{\pi}(x_2) = 0.5 \)

- Let \( x_i \) and \( \neg x_i \) takes maximum value in \( \ell_i \) 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: \( \text{OptVal}(F) \in C_N \) for \( N \in 2^{\text{poly}(\text{size}(F))} \).

\( C_N \): Farey Sequence of order \( N \). Fractions of the form \( A/B \), where \( 1 \leq A, B \leq N \) and \( \gcd(A, B) = 1 \).
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,

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

$$\hat{\pi}(F) \leq \frac{1}{4^{m-r}}$$
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 $F \rightarrow F'$: $C_i \rightarrow (C_i \lor y_i) \land (\neg y_i)$ for each $i$
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 $F \rightarrow F'$: $C_i \rightarrow (C_i \lor y_i) \land (\neg y_i)$ for each $i$

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

- We can give an interpretation $\pi$ 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 $m + r$
MaxSAT ≤ 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

\[ \text{FP}^{\text{NP}[\log]} \leq \text{OptVal} \leq \text{FP}^{\text{NP}} \]

Can we close the gap?

Two possibilities

- **OptVal ∈ FP^{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

\[ \text{FP}^{NP[log]} \leq \text{OptVal} \leq \text{FP}^{NP} \]

Can we close the gap?

Two possibilities

- **OptVal ∈ FP^{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 FP^{NP}-hard**
  - Well, a natural problem that’s complete for FP^{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](http://www.cs.toronto.edu/~meel/talks.html)
\( L_{\text{opt}} \) 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.
- \( \text{optSemVal} \) of a proof tree is of the form \( \left( \frac{a}{a+b} \right)^a \cdot \left( \frac{b}{a+b} \right)^b \).
- \( \text{optSemVal}(\phi) \) is the maximum over \( \text{optSemVal}(T) \) over all proof trees \( T \).
- NP Algorithm: Guess a proof tree \( T \) and compute its \( \text{optSemVal} \).

Algorithm

- Perform Binary search using \( L_{\text{opt}} \) 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.