# MaxSAT for xAI

My thoughts on Slides 6 [Email from Srivas: May $6^{th}$]:

- "Will MaxSAT, by default, produce a satisfying instance that satisfies as few of the soft clauses?" — I believe there might be some confusion regarding MaxSAT. We aim to query with hard constraints and uniform weight soft constraints. The solver's objective is to provide a satisfying assignment that meets all the hard constraints while maximizing the weights of the satisfiable soft constraints. In simpler terms, the solver aims to minimize the number of unsatisfied soft constraints. This formalization is known as the Weighted Partial MaxSAT problem. While we assume uniform weight for soft constraints, they can be prioritized using weights.
- Before delving into the query, let us recall the given condition $\forall x \in F \ {}_{\wedge i}(x_i = v_i) \rightarrow N(x) = c$, that is, given input is classified in class $c$. Now, let's examine the query:

$$\exists x \in F.(Soft :_{\wedge i} (x_i = v_i)) \wedge (Hard : N(x) \neq c)$$

Consider a scenario where the solver manages to satisfy all constraints, including both hard and soft ones. In such a case, for the same input $\forall x \in F \ {}_{\wedge i}(x_i = v_i)$, the network would classify it into a class different from $c$. This presents a contradiction since an input should ideally be classified into exactly one class. Therefore, this scenario is not viable, necessitating the solver to discard (unsatisfy) some soft constraints.

Now, let's assume that the solver provides a satisfying assignment, denoted as sigma:

$\sigma := \forall x \in U \ {}_{\wedge i}(x_i \neq v_i) \wedge \forall x \in F \setminus U \ {}_{\wedge i}(x_i = v_i)$. Here, $U$ is a subset of $F$. This signifies that the solver had to discard constraints corresponding to features in set $U$. Now, the question arises: what does set $U$ represent? Does it denote the set of relevant features? Does the set $F \setminus U$ represent the non-essential features?

My intuition suggests that $U$ will include at least one relevant feature, as changing their values would lead to a change in class. Though I lack formal arguments, consider the following example for illustration:

Let's consider a scenario where we have three features: $\{x_1, x_2, x_3\} \in F$, and we're dealing with a binary classification problem (I know that we don't have access to samples, this is just to understand the MaxSAT query results).

In this example, $x_1$ and $x_3$ are relevant features, and they must take values 0 and 1 respectively to classify into the $c_1$ class. Have a look at Table 1.

Returning to our setting, let's assume the given input is $I : \langle x_1 = 0, x_2 = 1, x_3 = 1 \rangle$, which classifies into class $c_1$. Our task is to find the relevant features using MaxSAT.

Our MaxSAT query becomes:

$Soft : ((x_1 = 0) \wedge (x_2 = 1) \wedge (x_3 = 1)) \wedge (Hard : N(x) \neq c_1)$

| $x_1$ | $x_2$ | $x_3$ | Classfication |
|---|---|---|---|
| 0 | 0 | 0 | $c_2$ |
| 0 | 0 | 1 | $c_1$ |
| 0 | 1 | 1 | $c_1$ |
| 1 | 0 | 1 | $c_2$ |

Table 1: An Example

Given that the penalty for dropping each soft constraint is 1 (weight of each soft constraint is 1), the solver will aim to minimize the penalty. Now, the solver needs to drop either $(x_1 = 0, x_2 = 1)$ or $(x_2 = 1, x_3 = 1)$ (both have an equal penalty of 2) to achieve a satisfying assignment. Hence, set $U$ is either $\{x_1, x_2\}$ or $\{x_2, x_3\}$. Recall that the relevant feature set (denoted as set $E$ in slides) is $\{x_1, x_3\}$. Each such assignment will include either $x_1$ or $x_3$.

How about if we change our query to this (which was my original proposal)?

$$\exists x \in F.(Soft :_{\wedge i} (x_i \neq v_i)) \wedge (Hard : N(x) = c)$$

Considering the same input setting: $I : \langle x_1 = 0, x_2 = 1, x_3 = 1 \rangle$, and $I$ classifies in class $c_1$.

Our new MaxSAT query becomes:

$Soft : ((x_1 \neq 0) \wedge (x_2 \neq 1) \wedge (x_3 \neq 1)) \wedge (Hard : N(x) = c_1)$

In this case, the solver must discard the constraints $(x_1 \neq 0)$ and $(x_3 \neq 1)$ in order to classify into class $c_1$. Consequently, the set $U := \{x_1, x_3\}$ is the set of relevant features. However, the question remains: can we formally demonstrate that set $U$ consistently represents the set of relevant features?

- **About Heuistics:** Exact MaxSAT solvers typically employ branch and bound (BnB) algorithms, often augmented with various heuristics to enhance their efficiency. For detailed insights, we can refer to the Handbook of Satisfiability, Chapter 23 – "Branch and Bound (BnB) scheme for solving the minimization version of MaxSAT: Given a MaxSAT instance, BnB explores the search tree that reprerents the space of all possible assignments for instance in a depth-first manner. At every node, BnB compares the upper bound (UB), which is the best solution found so far for a complete assignment, with the lower bound (LB), which is the sum of the number of clauses which are unsatisfied by the current partial assignment plus an underestimation of the number of clauses that will become unsatisfied if the current partial assignment is completed. If LB $\geq$ UB, the algorithm prunes the subtree below the current node and backtracks chronologically to a higher level in the search tree. If LB $<$ UB, the algorithm tries to find a better solution by extending the current partial assignment by instantiating one more variable. The optimal number of unsatisfied clauses in the input MaxSAT instance is the value that UB takes after exploring the entire search tree."

- Certainly, the worst-case time complexity of MaxSAT is exponential. However, similar to SAT, solvers have advanced significantly – MaxSAT competition result https://maxsat-evaluations.github.io/2023/descriptions.html. I believe we should be able to scale effectively, but we need to conduct basic experiments to understand.
- Similar to SAT solving techniques, we can guide the search towards a satisfying assignment by providing predicates or bounding boxes, which could potentially aid the solver. We can also assign higher weights (penalties) to constraints within the bounding box, which may assist also the solver. However, empirical experiments are necessary to gain a better understanding.