Tuesday 8th October, 2019

"Synthesize program safety spec by learning from membership"?

Let $X = \{x_1, ..., x_i\}$ be the named program variables. Let $N = \{n_1, ..., n_i\}$ be the numeric variables of finite domain. Let atomic predicates involving x_i, x_j, n_i be of the form $x_i <= x_j \mid x_i <= n_i \mid n_i <= x_i$. Let B representing a set of safety-violating program states be a formula of atomic predicates involving X, N. The safety problem is to check $SAT(B)$.

Let B be not explicitly known. Suppose an honest oracle O that answers membership query is given; this can be a human teacher who knows if some certain state $B(X_k, N_k)$ realized by concrete valuations of X_k, N_k is undesirable but at the same time cannot give a full account of B. (Impossible?)

The goal is to check $SAT(B)$ by learning the specification of B from O.

Thing 1. *Can we know if* $SAT(B)$ can *be answered efficiently? (Property Testing)*

If so, can we learn B *efficiently in such a way that allows* $SAT(B)$ *be checked efficiently? (Learnability)*

For example, can we test if B can *be represented in horn, and, if so, what is it in horn?*

If not, can we answer $SAT(B)$ *without knowing* B*? Perhaps by synthesizing a "good enough" hypothesis* H *such that* $SAT(H)$ *can be answered efficiently and that (Approximation)* $\|B - H\| <= \epsilon$ *or (Overapproximation)* $H \to B$*?*

Thing 2 (Property Testing). *What are the characterizing properties of the concept class* E *such that* $SAT(e)$ *can be checked efficiently for all* $e \in E$*? Can these properties be checked efficiently?*

If such E *cannot exist, what are the "useful" classes* $E_1, ..E_n$ *that can cover "most" practical cases?*

For example, what are the properties of the class of horns H*, maybe in terms of their Fourier spectrum? And how to check if* $B \in H$ *efficiently?*

Thing 3 (Learnability). *If we know there is an representation of* B *with which* $SAT(B)$ can *be answered efficiently, how to learn that representation efficiently?*