Learning Conjunctions with Equivalence and Membership Queries

Dimitris Diochnos

Spring 2019

Monotone Conjunctions

Membership Queries. We can learn monotone conjunctions using membership queries. The all 1s vector $x_{one} = \langle 1111\ldots1 \rangle$ is a true (positive) point for every monotone conjunction. So, one idea is to consider the sequence of instances x_i, with $i \in \{1, 2, \ldots, n\}$ that are obtained from the vector x_{one} but in coordinate i we have a 0. Therefore, if the membership query returns 1 for x_i, this means that x_i is a true point of the target concept c, which in turn implies that the i-th variable is not relevant for the target (since it was the only 0 in x_i). When the membership query returns a 0, this means that the i-th coordinate indeed corresponds to a relevant variable, since now the unique 0 in the truth assignment x_i implies that x_i is a negative point for c and therefore the i-th variable is falsified. Hence, after exactly n membership queries we can determine exactly which variables appear in the target concept c.

Equivalence Queries. We can learn monotone conjunctions using equivalence queries alone. In this case we initialize the hypothesis h to be the monotone conjunction of n Boolean variables and begin making equivalence queries; that is, $h = x_1 \wedge x_2 \wedge x_3 \wedge \ldots \wedge x_{n-1} \wedge x_n$.

- The first equivalence query is either correct and we are done, or it has to return a counterexample $x \in h \triangle c$ and therefore it holds that $h(x) = 0$ and $c(x) = 1$. Note that x can not be the all 1s vector x_{one}, since x_{one} is a true point for every monotone conjunction. Therefore, all the 0s that appear in x correspond to variables that are not relevant to the target concept c and for this reason we can update our hypothesis h and remove the said variables.
- We repeat asking equivalence queries with the updated hypothesis h until we reach the target function. Note that during this learning process we only get *positive* counterexamples. Therefore, after at most $n + 1$ equivalence queries we will receive a positive answer from the equivalence query oracle and we will thus have identified the target concept precisely.

Not Necessarily Monotone Conjunctions

Equivalence Queries. The algorithm for learning monotone conjunctions with equivalence queries can handle conjunctions in general with a small modification.

- We first query for the identically false conjunction; e.g., $h_0 = x_1 \wedge \overline{x_1}$.
- The above query is either correct and we are done, or a positive counterexample x is returned since h_0 is the identically false function. But then x dictates the orientation of the variables in the target conjunction. For example, if the i-th bit in x is a 0, then this implies that either variable x_i does not occur at all, or it occurs in the form of $\overline{x_i}$; it can not be the case that x_i appears as is in the target concept c.

- So, from the previous step we have the starting hypothesis h_1, which is the full conjunction that satisfies the counterexample x that was returned. From this point and on we repeat the process that we followed earlier until we determine the target concept precisely.

Membership Queries. The algorithm for membership queries does not generalize to general conjunctions. The reason is that there is not one vector that is positive for all the functions in the class, similar to x_{one} which was positive for all monotone conjunctions. In fact, if all membership queries are answered negatively, then as long as we did not ask all 2^n truth assignments it is still possible that the target concept is satisfied by one of the unasked vectors or is identically false. Therefore we get a negative result: in order to learn an n-variable conjunction, one has to query all 2^n rows of the truth table in the worst case.