LEARNING DECISION LISTS IN THE INFINITE ATTRIBUTE MISTAKE BOUNDED LEARNING MODEL

Part 1. Notation

See colt ref for information about notation.

Representation dimension (f(input size)) or number of vars in x: n. r: number of relevant vars.

Let the oracle calls used by $p(r, |c|)h(n)$.

Part 2. Problem statements

1. IMB learning of DL

1.1. **Importance.** Learning DL attribute efficiently (ae) could be of practical importance.

Ae learning DL's will give us more understanding about ae-learning in general.

2. Subexponential IMB learning of DL

2.1. **Importance.** Could be a step towards poly time IMB learning.

3. ae learn DL's in MB

3.1. **Importance.** This will imply solution to the IMB learning problem due to the sequential / mapping learning algs construction.

4. ae learn DL's in UPAC

Believed to be hard for DNF's.

4.1. **Importance.** These could indicate whether learning DL's is truly easier than learning DNF's.

How would you learn DL's using MQ's?

4.2. **Problem with the goal.** ae learning in U may not make much sense: length of dl and term length of DNF restricted to $\log(1/\epsilon)$; whereas from VCD lower bound you expect to learn in $\Omega(1/\epsilon)$. But, the VCD lower bound may not hold for U.

4.3. **Learn dl's using EQ.**

4.3.1. *Importance.* One could then try to fiddle with the eq alg to make it ae.

4.3.2. *Non ae alg.* Can make eq alg to learn DL from the mb alg to learn dl.

5. Learn dl's using MQ

5.1. **Importance.** One could then try to get rid of the mq's. Not very promising.

6. Learn DNF's in U using ae DL alg

6.1. Importance. Could indicate that learning DL's efficiently is hard

6.2. Failed attempts. Can't use Bshouty's DNF to augmented dec tree to t-dl alg to show that ae learning dl's implies DNF learnability: Feature expansion based t-dl alg is always non ae.

6.3. ae learn PARITY using ae DL alg.

6.3.1. *Importance.* Implies learning PARITY with noise which implies learning DNF in U.

Part 3. Known facts

7. Monotone DL can be learnt in IMB

They are merely conjunctions and disjunctions.

8. Known MB algs to learn DL's

Best known $p(r, |c|)h(n) = O(nk)$ by the literals in bag algorithm. $r(\log(n/r)) \leq VCD \leq r \log n$. So, big gap between upper and lower bound.

8.1. Non poly time ae algs. Sequential halving alg. Klivans and Servedio winnow alg (take a look.).

k-CNF and k-DNF learnable using winnow with time per trial $O(n^k)$. Can use this to learn DL's in time per trial $O(n^r)$.

Part 4. Strategies for faster algorithms

9. Winnowing

Use some form of winnowing.

9.1. Conversion to halfspace. Decn list with k vars $\{x_i\}$, outputs $r(x_i)$ can be writ as halfspace: $sign(\sum 2^{k-i} x_i o(x_i))$. Can we use winnow to set these weights?

10. Occam with Approximate set cover

Try to find an algorithm which will grow the hypothesis decision list efficiently, as if solving a set cover problem. **Initial attempt's result**: Discovered that the set-cover logic cannot be used directly in the way it is used in the case of disjunctions.

11. Vague thoughts

DL's are special types of r-term r-DNF's. We must exploit this structure some how.

How to efficiently identify the first var in a DL?

Part 5. Attempts at showing hardness

If you could learn DL's ae, could you learn DNF's in poly time? Seems to be no obvious way.