Hardness Results for Learning DNF

Lev Reyzin
Clique Talk, 2007
The Papers this Talk Covers

- Alekhnovich, Braverman, Feldman, Klivans, Pitassi, New Results on Hardness of Proper Learning (FOCS 2004, JCSS 2005) [ABFKP]
  If NP \neq RP, then DNF are not properly PAC learnable

- Feldman, Hardness of Approximate Two-level Logic Minimization and PAC Learning with Membership Queries (STOC 2006) [Feldman]
  [ABFKP] holds true even when the learner has access to membership queries
The Model and Definitions

- **PAC learning [Valiant ’84]**
  - Algorithm A (efficiently) PAC learns class C of functions \( \{0,1\}^n \rightarrow \{0,1\} \) if for every \( \epsilon>0, \delta>0, n, c \) in C, and distribution \( D_n \), A outputs a hypothesis \( h \) from class \( H \) that \( \epsilon \)-approximates \( c \) with probability \( 1-\delta \) and runs in time \( \text{poly}(n,1/\epsilon,1/\delta,|c|) \).
  - If \( H=C \), then A is a proper PAC learning algorithm
  - Given an example oracle that upon request returns example \( (x,c(x)) \), where \( x \) is chosen randomly w.r.t. \( D \)

- **DNF formulas and Threshold Functions**
  - A DNF formula is equal to ORs of ANDs, ie \( (x_1 \land x_2 \land x_4) \lor (x_5 \land x_1) \)
  - A k-term DNF is a DNF formula equal to the OR of k ANDs
  - A threshold function is a function \( f = \text{sign}(\sum (\alpha_i x_i) - \theta) \) where all \( \alpha_i \) and \( \theta \) are integers.
In his seminal paper introducing PAC learning, Valiant ['84] posed the question whether DNF are properly PAC learnable.

Pitt and Valiant ['88] showed that it is NP hard to learn k-term DNF by k-term DNF.

On the other hand we can PAC learn DNF in sub-exponential time. [Bshouty ’96]

This result answers Valiant’s long-open question.
A Quick Warm-Up

- **k–Colorable hypergraphs**
  - A **k–coloring** of a hypergraph means finding a mapping from the vertices to \{1, ..., k\} s.t. no edge has all of its vertices assigned the same color.

- **Theorem [Pitt, Valiant ’88]**
  Coloring a k–colorable hypergraph \( H = (V, E) \) using \( L \) colors reduces to learning k–term DNF formulae by outputting \( L \–term \ DNF \) formulae.
An Illustration of the Reduction

\[ t_{\text{blue}} = (x_2 \land x_4 \land x_3 \land x_5) \]
\[ t_{\text{red}} = (x_1 \land x_3 \land x_5 \land x_6) \]
\[ t_{\text{green}} = (x_1 \land x_2 \land x_4 \land x_6) \]
\[ h = t_{\text{blue}} \lor t_{\text{red}} \lor t_{\text{green}} \]

\[ h = t_1 \lor t_2 \lor t_3 \]

\[ t_1 = (x_2 \land x_3 \land x_4 \land x_5) \]
\[ t_2 = (x_1 \land x_4 \land x_5 \land x_6) \]
\[ t_3 = (x_1 \land x_2 \land x_3 \land x_6) \]
The Reduction

**Proof of Thm [PV]** Coloring a $k$–colorable hypergraph $H=(V,E)$ using $L$ colors reduces to learning $k$–term DNF by outputting $L$–term DNF

- let $H(V,E)$ be any $k$–colorable hypergraph on $n$ vertices
- make set $S$: for each $v \in V$, $(a(v), +)$ and $e \in E$, $(a(e), -)$
  - $a(v_i) =$ length $n$ vector w/ 0 at position $i$ and 1 elsewhere
  - $a(e) = \Lambda_{v \in e}a(v)$ bitwise
- any $k$ coloring of $H \leftrightarrow$ DNF consistent w/ examples
  - let $\chi$ be a $k$–coloring of $H$, for every color $c$, let $t_c = \bigwedge_{\chi(v_i) \neq c} x_i$
  - we set $h = t_1 \lor t_2 \lor \ldots \lor t_k$
  - for each vertex example $a(v_i)$, $t_{\chi(v_i)}(a(v_i)) = 1$, and hence $h(a(v_i)) = 1$
  - for any edge example $a(e)$, $h$ will not satisfy $a(e)$
  
  \[ h = t_1 \lor t_2 \lor \ldots \lor t_L \] be a DNF consistent with the given examples
  - for each vertex, we define $\chi(v_i) = c$ for least $c$ s.t. $a(v)$ is satisfied by $t_c$
  - this defines a mapping from vertices to colors
  - take $e \in E$, assume that all its vertices are colored in $c$, $\forall v \in e, t_c(a(v)) = 1$
  
\[ t_c(a(e)) = t_c(\bigwedge_{v \in e} a(v)) = \bigwedge_{v \in e} t_c(a(v)) = 1 \]
Theorem [PV] Coloring a $k$–colorable hypergraph $H = (V, E)$ using $L$ colors reduces to learning $k$–term DNF formulae by outputting $L$–term DNF formulae

Theorem [Dinur, Regev, Smyth ’02] It is NP Hard to $k$–color a 2–colorable 3–uniform hypergraph for any constant $k$

Theorem [ABFKP] Assuming NP $\neq$ RP, there is no polynomial–time algorithm for learning 2–term DNF formulae by $k$–term DNF formulae for any constant $k$
Make examples from graph

- Have small CNF consistent with examples
  - If CNF is learnable we can approximate chromatic number
  - But we (probably) can’t approximate chromatic number

- No small AND of threshold consistent with examples
  - If DNF is learnable we can approximate chromatic number
  - So unless \( \text{NP} = \text{RP} \) we can’t learn DNF

Small \( X \)

Large \( X \)
For some $r$, examples from $\{0,1\}^{n \times r} = (\{0,1\}^n)^r$

Let $G(V,E)$ be a graph on $n$ vertices, $m$ edges
define vectors $z$:  

- $z(v_2) = (0,1,0,0,0,0)$
- $z(v_5) = (0,0,0,0,1,0)$
- $z(v_6) = (0,0,0,0,0,1)$
- $z(\varepsilon_{5,6}) = (0,0,0,0,1,1)$

The Distribution $D$

- for each vector $(v_1, \ldots, v_r) \in V^r$ associate a negative example $(z(v_1), \ldots, z(v_r), -)$. Giving $n^r$ negative examples $(S^-)$
- for each choice of $k_1, k_2$ s.t. $1 \leq k_1 \neq k_2 \leq r$, $e \in E$, and $v_i \in V$ for each $i \neq k_1, k_2$ we associate a positive example $(z(v_1), \ldots, z(e), z(v_{k_1+1}), \ldots, 0, z(v_{k_2+1}), \ldots, z(v_r), +)$, giving a total of $r(r-1)|E|n^{r-2}$ positive examples $(S^+)$
- $D$ sets probability of each negative example to be $1/2n^r$ and of a positive example $1/2r(r-1)|E|n^{r-2}$

$S = S^+ \cap S^-$
Graph Coloring → CNF (Small $\chi$)

- We’ll show $\chi(G)$ is “small” → exists “small” CNF consistent with the examples

**Lemma** If $\chi(G) \leq n^\gamma$, then $\exists$ a CNF of size $n^{\gamma r}$ consistent with the examples.

- suppose $V = \bigcup_{i=1}^{\chi} I_i$, where $I_i$ are independent sets
- define the CNF formula $f(x_1, ..., x_n) = \bigwedge_{i=1}^{\chi} \bigvee_{j \notin I_i} x_j$
- define a formula on $r \cdot n$ vars, consistent w/ the learning problem: $F((x_1, ..., x_n), ..., (x_1^r, ..., x_n^r)) = \bigvee_{k=1}^{r} f(x_1^k, ..., x_n^k) = \bigvee_{k=1}^{r} \bigwedge_{i=1}^{\chi} \bigvee_{j \notin I_i} x_j^k$
  - each vertex will fail on the independent set it’s a part of
  - each edge has one “foot” in two independent sets
- $F$ is a disjunction of $r$ CNF w/ at most $\chi(G)$ clauses. Expanding the formula yields a CNF with $\chi(G)^r = n^{\gamma r}$ clauses, satisfying the lemma
The Case of Large $\chi$

**Theorem [ABFKP]** Let $G$ be a graph such that $\chi(G) \geq n^{1-\gamma}$. Let $F = \bigwedge_{i=1}^l h_i$, $l < \frac{1}{2^{2\gamma r}} \left( \frac{\chi-1}{\log n} \right)$. Then $F$ has error at least $1/n^{2\gamma r + 4}$ with respect to $D$.

- **in other words**, we assume that $\chi(G) \geq n^{1-\gamma}$, and we prove no “small” AND–of–thresh. formula gives a good approximation to the learning problem.

**Covering Lemma [Linial, Vazirani ’89; Feige ’95]** One needs at least $((\chi-1)/\ln(n))^r$ products of the form $I_1 \times I_2 \times \ldots \times I_r$ to cover $V^r = V \times \ldots \times V$.

- **we will show** that any $h_k \in F$ correctly classifies few negative examples that lie outside a particular product of independent sets.
On Independent Sets

- Remember $F = \bigwedge_{i=1}^{l} h_i$, $l < \frac{1}{2\chi r} \left( \frac{\chi - 1}{\log n} \right)^r$ and fix an $h_k \in F$
- let $h_k = \sum_{i=1}^{r} \sum_{j=1}^{n} \alpha_j x_j \geq \beta$
- for each $i \leq r$ the $i$-coefficients are $\alpha_j$ for $j \leq n$
- for each $i \leq r$ let $I_i$ be the set of all $j \leq n$ s.t. there is no edge $(k,j) \in E$ with $\alpha_k < \alpha_j$
  - this orders all $i$-coefficients in nondecreasing order and takes independent coefficients in that order
  - $I_i$ is an independent set in $G$
- let $S_1^k = V \times I_2 \times I_r$, $S_2^k = I_1 \times V \times I_3 \times I_r$ and so on
- let $S_k = \bigcup_{i=1}^{r} S_i^k$
  - we will show $h_k$ either misclassifies many positive examples or most negative ones outside $S^k$
**Forced Misclassification**

- **Error Lemma** Fix $h_k, l_1, ..., l_r$, and $S_k$ as before. Let $N$ be the number of negative examples outside of $\bigcup_{k=1}^{l} S^k$ that $h_k$ classifies correctly. Then the number of positive examples that $h_k$ (and therefore $\Lambda h_k$) misclassifies is at least $N/2n$.
  - The intuition is that one threshold function cannot classify too many examples correctly. We can produce a mapping from correctly classified examples to incorrectly classified ones.

For each $h_k$, Let $\text{In}_k$ be the correctly classified negative examples in $S^k$. Let $\text{Out}_k$ be the remaining correctly classified negative examples.
**Error Lemma** Fix $h_k, I_1, \ldots, I_r$, and $S_k$ as before. Let $N$ be the number of negative examples outside of $\bigcup_{k=1}^r S^k$ that $h_k$ classifies correctly. Then the number of positive examples that $h_k$ misclassifies is at least $N/2n$.

- Let $\alpha = z(j_1), \ldots, z(j_r)$ be a negative example s.t. $\alpha$ is not in $S^k$.
  - therefore $h_k(\alpha) < \beta$
- Since $\alpha$ is not in $S^k$, $\exists$ two $j_i$’s, ie $j_1$ and $j_2$ s.t. $j_1 \notin I_1$ and $j_2 \notin I_2$
  - this implies $\exists$ a vertex $k_1$ in $I_1$ s.t. edge $(j_1, k_1) \in E_1$ (for $k_2$ resp.)
- by how we chose $I_1, I_2$ it follows $\alpha^1_{k_1} \leq \alpha^1_{j_1}$ and $\alpha^2_{k_2} \leq \alpha^2_{j_2}$
  - either a) $\alpha^1_{j_1} \leq \alpha^1_{j_2}$ or b) $\alpha^1_{j_2} < \alpha^1_{j_1}$
- if a) then $\alpha^1_{j_1} + \alpha^1_{k_1} + \alpha^3_{j_3} + \ldots + \alpha^r_{j_r} \leq \beta$
  - the + example $\alpha' = (z(j_1, k_1), 0, z(j_3), \ldots, z(j_r))$ is misclassified by $\Lambda h_k$
- if b) then $h_k$ (and $\Lambda h_k$) and + ex. $\alpha' = (0, z(j_2, k_2), z(j_3), \ldots, z(j_r))$
  this gives us a mapping from correctly classified negative ex. to misclassified + ex. Since each + ex. is mapped onto by at most $2n$ negative examples, this finishes the proof of the lemma.
Lemma Let $S^k$, $k \leq l$ be defined as before. If

$$l \leq \frac{1}{2} \chi r \left( \frac{\chi - 1}{\ln n} \right)^r$$

then

$$n^r - \left| \bigcup_{k=1}^{l} S^k \right| \geq \frac{1}{2} \left( \frac{\chi - 1}{\ln n} \right)^r$$

Proof follows:

◦ Assume to the contrary. We would then have a collection of

$$l \chi r \leq \frac{1}{2} \left( \frac{\chi - 1}{\ln n} \right)^r$$

products of independent sets, which would cover all but

$$m \leq \frac{1}{2} \left( \frac{\chi - 1}{\ln n} \right)^r$$

points of $V^r$.

◦ By adding $m$ singletons (which are ind sets) we get a cover of $V^r$ by

$$l \chi r + m \leq \left( \frac{\chi - 1}{\ln n} \right)^r$$

independent sets, contradicting the covering lemma.

We can now analyze the overall error wrt $D$

◦ Let $F = \bigwedge_{i=1}^{l} h_i$, $l < \frac{1}{2} \chi r \left( \frac{\chi - 1}{\log n} \right)^r$ $h_i$’s are threshold formulas

◦ Let $R = \frac{1}{4} \left( \frac{\chi - 1}{\ln n} \right)^r$ we split into two cases

  • when $\left| \bigcup_{k}^{l} Out_k \right| \geq R$ by the Error Lemma, $F$ misclassifies $\geq R/2n$ positive examples. so the probability of error wrt $D$ is at least $1/n^{2r\gamma + 4}$

  • when $\left| \bigcup_{k}^{l} Out_k \right| < R$ by the Lemma above, $F$ misclassifies at least

$$\frac{1}{2} \left( \frac{\chi - 1}{\ln n} \right)^r - R$$

negative examples. This makes the error wrt $D$ at least $R/2nr$, which is at least $1/n^{2r\gamma + 4}$ for large $n$. □
The Error Calculation

- Let \( R = \frac{1}{4} \left( \frac{x-1}{\ln n} \right)^r \), we split into two cases:
  - when \( \bigcup_k^{r'} \text{Out}_k \geq R \) by the Error Lemma, \( F \) misclassifies \( \geq R/2n \) positive examples.
    - Thus the probability of error wrt \( D \) is \( R/(4nr(r-1)|E|n^{r-2}) \), which is at least:
      \[
      \frac{R}{n^{r+4}} = \frac{1}{4} \left( \frac{x-1}{\ln n} \right)^r \geq \frac{1}{4} \left( \frac{1-\gamma-1}{\ln n} \right)^r > \frac{1}{n^{r+4}} = n^{-2\gamma-4} = \frac{1}{n^{2\gamma+4}}
      \]
      - so the probability of error wrt \( D \) is at least \( 1/n^{2\gamma+4} \)
  - when \( \bigcup_k^{r'} \text{Out}_k < R \) by the Lemma on previous slide, \( F \) misclassifies at least \( \frac{1}{2} \left( \frac{x-1}{\ln n} \right)^r - R = R \) negative examples.
    - This makes the error wrt \( D \) at least \( R/2nr \), which is at least \( 1/n^{2\gamma+4} \) for large \( n \).
Approximating $\chi$ by Learning CNF

**Theorem [ABFKP]** If CNF is learnable by ANDs of thresholds in time $O(n^k s^k (1/\epsilon)^k)$ for $k>1$ then there exists a randomized algorithm for approximating $\chi$ of a graph within a factor of $n^{1-1/(10k)}$ in time $O(n^{9k})$.

**The Algorithm**
- Set $\epsilon = 1/(n^6)$ and $r=10k$. Let $G$ be the graph and $D$ be the distribution induced by $G$.
- Run learning algorithm wrt $D$. If it does not terminate within $n^{9k}$ steps, say “$\chi > n^{1-1/(10k)}$”
- Else let $h$ be hypothesis and $\epsilon_h$ its error wrt $D$
- If $\epsilon_h < \epsilon$ say “$\chi < n^{1/(10k)}$” else say “$\chi > n^{1-1/(10k)}$”
Why the Algorithm Works

The Algorithm
- Set $\epsilon = 1/(n^6)$ and $r=10k$. Let $G$ be the graph and $D$ be the induced distribution. Run learning algorithm wrt $D$. If it does not stop within $n^{9k+1}$ steps, say “$\chi \geq n^{1-1/(10k)}$”. Else let $\epsilon_h$ be the error of $h$ wrt $D$. if $\epsilon_h < \epsilon$ say “$\chi \leq n^{1/(10k)}$” else say “$\chi \geq n^{1-1/(10k)}$”

Correctness
- If $\chi \leq n^{1/(10k)}$, by “small $\chi$ Lemma,” $s \leq n^{1/10k*10k}$, and the number of variables is $r \cdot n \leq n^2$. So w.p. $\frac{3}{4}$ the running time is $O((10kn)^k(n)^kn^6k) \leq O(n^{8k}) < n^{9k}$ for large $n$. So the Alg. outputs “$\chi \leq n^{1/(10k)}$” w.p. $\geq \frac{3}{4}$.
- If $\chi \geq n^{1-1/(10k)}$, by “large $\chi$ Lemma”, the algorithm must contain at least $1/(2 \chi r)^*((\chi-1)/\ln n)^r$ terms to have error $< \epsilon$. In this case the running time is at least $1/(2 \chi r)^*((\chi-1)/\ln n)^r \geq n^{9k}$ for large $n$. 
Old Theorem If CNF is learnable by ANDs of thresholds in time $O(n^{ks^k(1/\varepsilon)^k})$ for $k>1$ then there exists a randomized algorithm for approximating $\chi$ of a graph within a factor of $n^{1-1/(10k)}$ in time $O(n^{9k})$.

New Theorem If DNF is learnable by ORs of thresholds in time $O(n^{ks^k(1/\varepsilon)^k})$ for $k>1$ then there exists a randomized algorithm for approximating $\chi$ of a graph within a factor of $n^{1-1/(10)}$ in time $O(n^{9k})$. 
From Previous Slide: If DNF is learnable by ORs of thresholds in time $O(n^k s^k (1/\varepsilon)^k)$ for $k > 1$ then there exists a randomized algorithm for approximating $\chi$ of a graph within a factor of $n^{1-1/(10k)}$ in time $O(n^{9k})$.

Theorem [Feige and Kilian ’96] Let $\varepsilon > 0$ be a constant. Assume there exits an algorithm that approximates the chromatic number of a graph on $n$ vertices to a factor of $n^{1-\varepsilon}$ in $\text{RPTIME}(t(n))$, then $\text{NP} \subseteq \text{RPTIME}(t(n^\alpha))$ for some $\alpha \geq 1$.

So we get $\text{NP} \subseteq \text{RPTIME}(n^{O(1)}) \subseteq \text{RP}$

So if $\text{NP} \neq \text{RP}$, then DNF are not properly PAC learnable [ABFKP]
In introducing PAC learning, Valiant [’84] also posed the question whether DNF are properly PAC learnable with membership queries.

- monotone DNF are strongly PAC learnable with MQs [Valiant ’84].
- If non-uniform 1-way functions exist, MQs don’t help in PAC learning DNF [Angluin and Kharitonov ’95]
  - can’t combine this with [ABFKP] to get [F]
- this result answers Valiant’s other long open question
Theorem [Feldman] If NP ≠ RP then there is no polynomial-time proper PAC learning algorithm for DNF expressions even when the learning algorithm has access to the membership oracle.

Membership Queries The learner is given access to a membership oracle that, given any point \( x \in X \), returns the value \( c(x) \).

Proof Idea define values on the target function \( f \) on the rest of the hypercube so that in the case of the “small” chromatic number, \( f \) can still be represented by a relatively “small” CNF formula. This allows us to answer queries to the membership oracle without any knowledge of a “small” coloring.
The Distribution $D$

- for each vector $(v_1, \ldots, v_r) \in V^r$ associate a negative example $(z(v_1), \ldots, z(v_r), -)$.
- for each choice of $k_1, k_2$ s.t. $1 \leq k_1 \neq k_2 \leq r$, $e \in E$, and $v_i \in V$ for each $i \neq k_1, k_2$ we associate a positive example $(z(v_1), \ldots, z(e), z(v_{k_1}+1), \ldots, 0, z(v_{k_2}+1), \ldots, z(v_r), +)$
- for the rest of the points on the hypercube

On the rest of the hypercube we define $f$ as follows: let $x = (x^1, \ldots, x^r)$ be a point not in $S^+ \cup S^-$ If for all $i$

- If $\forall i$, $x^i \in \{0\} \cup \{z(v) \mid v \in V\}$ then $f(x)=0$ 0–vertex points
- If $\exists i \leq r$, $j_1, j_2$ s.t. vertices with indices $j_1$ and $j_2$ are not connected by an edge in $G$ and $x^i_{j_1} = x^i_{j_2} = 1$, then $f(x)=0$ non–edge points
- Otherwise, let $f(x)=1$
The Case of Small $\chi$ Revisited

- **Lemma** If $\chi(g) \leq n^\lambda$, then there is a CNF formula of size at most $n^{r\lambda} + r|E|$ equal to $f$.

  - suppose $V = \bigcup_{i=1}^{\chi} I_i$, where $I_i$ are independent sets.
    - define the CNF formula $g(x_1, \ldots, x_n) = \bigwedge_{i=1}^{\chi} \bigvee_{j \notin I_i} x_j$
      - this formula rejects all points in $\{o\} \cup \{z(v) \mid v \in V\}$ and accepts all points in $\{z(e) \mid e \in E\}$
  - $F$ rejects all the points in $S^-$ and the 0–vertex points. $F((x_1^1, \ldots, x_n^1), \ldots, (x_1^r, \ldots, x_n^r)) = \bigvee_{k=1}^{r} f(x_1^k, \ldots, x_n^k) = \bigvee_{k=1}^{r} \bigwedge_{i=1}^{\chi} \bigvee_{j \notin I_i} x_j^k$
    - we can write $F$ in CNF like [ABFKP]
  - Define CNF formula $H$ on $r n$ variables that rejects all non–edge points. $H(x_1^1, \ldots, x_n^r) = \bigwedge_{k \leq r; (u, w) \notin E} (x_i^k \vee \overline{x_i^k})$
    - this has size $r|E|$
  - So $F \land H$ can be written as CNF satisfying the lemma
For the case of large $\chi$ we use the analysis in [ABFKP] since our definition of the target function on points of weight 0 does not make the task of finding an AND-of-thresholds formula with small error any easier or harder. (with a small correction)

This proves the Theorem [Feldman] If $\text{NP} \neq \text{RP}$ then there is no polynomial-time proper PAC learning algorithm for DNF expressions even when the learning algorithm has access to the membership oracle.
Open Questions & Future Directions

- Still open: can we (not-properly) PAC learn DNF?
- We know an OR-of-Thresholds cannot learn DNF. Can we prove that about more general classes?
  - this could be done by tweaking the “error lemma”
- This is the first NP hardness result for PAC learning with membership queries. Can we apply these techniques elsewhere?
- This result uses a lot of machinery. Maybe it can be simplified. ie g(n) function.