	CS 357 Algorithms	
Lecture 10: February 22, 2005		Lecturer: Adam Klivans
	Scribe: Irvin Hwang	

Hardness Results

10.1 Introduction

In this lecture we present some hardness results for learning, more specifically we show proper learning of the intersection of two half-spaces is NP-hard. We also show learning the intersection of two half-spaces when the output is k half-spaces, for a constant k, is NP-hard. We achieve these results using the consistency problem.

Definition 1 *Proper learning - An algorithm for learning a function $f \in F$ is a proper learning algorithm if the output hypothesis h is also in F (i.e. $h \in F$).*

For example, an algorithm for learning decision trees that outputs a decision tree would be a proper learning algorithm. The algorithm presented in class for learning decision trees by outputting decision lists would not be a proper learning algorithm.

Definition 2 *Consistency problem - The inputs for the consistency problem are a set of labelled examples and the output is a hypothesis consistent with the examples.*

The general idea behind the hardness results is to allow an instance of an NP-hard problem to become the input for the consistency problem (i.e. map an instance of the problem to a set of labelled examples). Thus, if there is a halfspace consistent with the examples then you can solve the NP-hard problem.

10.2 Properly Learning the Intersection of Two Half-Spaces is NP-hard

In this section we use the NP-hard problem of set-splitting to show learning the intersection of two half-spaces, when the output is the intersection of two half-spaces, is NP-hard. This result is due to [Blum, Rivest 1989].

10.2.1 Set-Splitting

The problem of set-splitting is defined as follows:

Given a set $S = \{s_1, s_2, \ldots, s_n\}$ and subsets $A_1, A_2, \ldots, A_m \subset S$

Output $S_1, S_2 \subset S$ such that $S_1 \bigcup S_2 = S$, $S_1 \bigcap S_2 = \emptyset$, and $\forall i : A_i \not\subset S_1$ and $A_i \not\subset S_2$ (i.e. split the set S with S_1 and S_2 such that no A_i is completely contained in either S_1 or S_2).

We want to map an instance of set-splitting, σ, to a set of labelled examples in order to show that learning the intersection of two half-spaces is NP-hard.

Labelled Examples $E(\sigma)$:

- $(0^n, +)$
- for each s_i, $\rho(s_i) = (0\ldots010\ldots0, -)$ where the 1-bit occurs in the ith position of the string.
- for each A_i, $A_i = \{s_{i1}, s_{i2}, \ldots, s_{ij}\} = (\rho(s_{i1}) + \rho(s_{i2}) + \ldots + \rho(s_{ij}), +)$ (i.e. a vector with j 1-bits)

Example 1 *Define an instance of split-setting, σ, to be...*

$$S = \{s_1, s_2, s_3\},\ A_1 = \{s_1, s_2\},\ A_2 = \{s_2, s_3\}$$

then the set of labelled examples, $E(\sigma)$, would be...

$$\{(000,+), (100,-) \sim s_1, (010,-) \sim s_2, (001,-) \sim s_3, (110,+) \sim A_1, (011,+) \sim A_2\}$$

10.2.2 The Consistency Problem, Set-Splitting, and Learning an Intersection of 2 Half-spaces

Theorem 1 *Learning an intersection of 2 half-spaces is NP-hard.*

This theorem is proved by first showing that given an instance of set-splitting, σ, there is an intersection of two half-spaces that is consistent with the examples (i.e. classifies $E(\sigma)$). Then we show given an intersection of two half-spaces consistent with $E(\sigma)$ we can find S_1 and S_2 that split S.

Proof:

Let $S = \{s_1, s_2, \ldots, s_n\}$ and $\{A_i\}_{i=1}^m$

Assume there is an instance of set-splitting, σ, where $\exists S_1, S_2$ such that $S_1 \bigcup S_2 = S, S_1 \bigcap S_2 = \emptyset, A_i \not\subset S_1$, and $A_i \not\subset S_2$.

Show there is an intersection of two half-spaces classifying $E(\sigma)$.

Let h_1, h_2 be the half-spaces and let x_i correspond to the ith bit of a labelled example...

$h_1 = \sum_{i=1}^n a_i x_i \geq -\frac{1}{2}$ where $a_i = \begin{cases} -1 & \text{if } s_i \in S_1 \\ n & \text{if } s_i \notin S_1 \end{cases}$

$h_2 = \sum_{i=1}^{n} b_i x_i \geq -\frac{1}{2}$ where $b_i = \begin{cases} -1 & \text{if } s_i \in S_2 \\ n & \text{if } s_i \notin S_2 \end{cases}$

h_1 and h_2 are consistent with $E(\sigma)$ because...

- For $(00\ldots0,+)$
  $h_1 = 0 \geq -\frac{1}{2}$ so its positive and $h_2 = 0 \geq -\frac{1}{2}$ is also positive
  therefore $h_1 \wedge h_2$ is positive.
- For $\rho(s_i) = (0\ldots010\ldots0,-)$
  Assuming, without loss of generality, $s_i \in S_1$ then $s_i \notin S_2$ since $S_1 \bigcap S_2 = \emptyset$. So...
  $h_1 = -1 \not\geq -\frac{1}{2}$ so its negative and $h_2 = n \geq -\frac{1}{2}$ is postive
  therefore $h_1 \wedge h_2$ is negative.
- For any $A_i = \{s_{i1},\ldots,s_{ik}\} = (\rho(s_{i1}) + \ldots + \rho(s_{ik}),-)$
  We know from our first assumptions about A_i that $A_i \not\subset S_1$ and $A_i \not\subset S_2$ so A_i must have at least one element from each set. This means at least one $a_i = n$ in h_1 and at least one $b_i = n$ in h_2. This value will then cancel out all the -1 terms in h_1 and h_2 so...
  $h_1 \geq 1 \geq -\frac{1}{2}$ so its positive and $h_2 \geq 1 \geq -\frac{1}{2}$ is also positive
  therefore $h_1 \wedge h_2$ is positive.

therefore $h_1 \wedge h_2$ is consistent with $E(\sigma)$.

Now we want to show the other direction.

Assume $h_1 \wedge h_2$ is consistent with $E(\sigma)$.

Show $\exists S_1, S_2$ such that $S_1 \bigcup S_2 = S, A_i \not\subset S_1$, and $A_i \not\subset S_2$.

Let S_1 be the set of points whose examples are classififed negative by h_1.
Let S_2 be the set of points whose examples are classififed negative by h_2.

We can show S_1 and S_2 solve set splitting.

- $S_1 \bigcup S_2 = S$
  For every s_i there exists a negative example and since $\bigcup_{i=1}^{n} s_i = S$ and all negative examples are covered by S_1 and S_2, $S_1 \bigcup S_2 = S$.
- $S_1 \bigcap S_2 = \emptyset$
  Suppose $S_1 \bigcap S_2$ was not empty and contained some s_k. That would mean both h_1 and h_2 classify s_k as negative, but this is impossible since both h_1 and h_2 would be positive if $s_k \in S_1$ and $s_k \in S_2$ by definition of h.
  therefore $S_1 \bigcap S_2 = \emptyset$.
- $A_i \not\subset S_1$, and $A_i \not\subset S_2$
  Suppose, without loss of generality, $A_i \subset S_1$
  By definition of h_1 and the assumption $A_i \subset S_1$, we know for any $x_i = 1$ the corresponding $a_i = -1$ for $h_1(A_i)$. So...
  $h_1(A_i) \leq -1 \not\geq -\frac{1}{2}$ so $h_1(A_i)$ is negative. But this is a contradiction because the labelled

examples corresponding to A_i must be positive by definition of $E(\sigma)$.
therefore $A_i \not\subset S_1$ and by the same argument $A_i \not\subset S_2$

therefore S_1 and S_2 solve set splitting.

We have now shown that learning an intersection of two half-spaces where the hypothesis is an intersection of two half-spaces solves the consistency problem where the input is an instance of set-splitting. Since set-splitting is known to be NP-hard, learning an intersection of two half-spaces must also be NP-hard. ■

10.3 Learning the Intersection of Two Half-spaces is NP-hard if the Output is k Half-spaces

In this section we use the NP-hard problem of hyper-graph coloring to show learning the intersection of two half-spaces, when the output is the intersection of k half-spaces for a constant k, is NP-hard. This result is due to [Alekhnovich, Braverman, Feldman, Klivans, Pitassi 2004].

10.3.1 Hyper-graph Coloring

We know that coloring a 3-uniform hyper-graph with k-colors, for a constant k, is NP-hard. In fact we know from [Dinur, Regev, and Smyth 2002] it is NP-hard even when the graph is 3-uniform and known to be 2-colorable. We will show that if you can learn the intersection of two half-spaces and output k halfspaces then you can color a hypergraph.

Definition 3 *Hyper-graph - A hyper-graph H is a set of vertices and edges, where an edge is a subset of the vertices.*

Definition 4 *c-uniform - A hyper-graph is c-uniform if each edge has size c.*

For example a 3-uniform hyper-graph has all edges of size 3.

Definition 5 *colorable - a hyper-graph is colorable if there is a mapping, $\chi(v) \rightarrow colors$, where v is the set of vertices in a hyper-graph, such that no edge is monochromatic.*

Coloring a hyper-graph consists of assigning colors to vertices such that no edge consists of vertices all of the same color (i.e. no edge is monochromatic).

We want to map an instance of hyper-graph coloring, $H = (V, E)$ where V is a set of vertices and E a set of edges, to a set of labelled examples.

Labelled Examples $E(H)$:

- $(0^n, +)$

- for each $v_i \in V$, let $\rho(v_i) = (0 \ldots 010 \ldots 0, -)$ where the 1-bit occurs in the ith position of the string.
- for each $e = \{v_i, v_j, v_k\} \in E$, let $\rho(e) = (\rho(v_i) + \rho(v_j) + \rho(v_k), +)$

10.3.2 The Consistency Problem, Hyper-graph Coloring, and Learning an Intersection of Half-spaces

Theorem 2 *Learning an intersection of 2 half-spaces when the output is k half-spaces, for a constant k is NP-hard.*

This theorem is proved in a similar manner to the earlier set-splitting problem. We first show that given an instance of hyper-graph coloring, H, there is an intersection of k half-spaces that is consistent with the examples (i.e. classifies $E(H)$). Then we show given an intersection of k half-spaces consistent with $E(H)$ we can find a coloring scheme for H.

Proof: Assume h is colorable by l colors.

Let h_j define a half-space and x_i be the ith bit of a labelled example.

$h_j = \sum_{i=1}^{n} w_{ji} x_i + \frac{1}{2}$ where $w_{ji} = \begin{cases} -1 & \text{if } \chi(v_i) = j \\ n & \text{otherwise} \end{cases}$

$h = \wedge_{j=1}^{l} h_j$ is the intersection of l half-spaces

h is consistent with $E(H)$ because...

- For $(00 \ldots 0, +)$
  $\forall j$ h_j is positive because $\sum_{i=1}^{n} w_{ji} x_i + \frac{1}{2} = \frac{1}{2}$ since $\forall i x_i = 0$
  therefore $h = \wedge_{j=1}^{l} h_j$ is positive since all h_j are positive.
- For $\rho(v_i) = (0 \ldots 010 \ldots 0, -)$
  Consider when v_i is colored b for some color b(i.e. $\chi(v_i) = b$). Then h_b is negative because $w_{bi} x_i = -1$ (since the bit corresponding to x_i is one and $\chi(v_i) = b$) and all other wx terms are zero since all other bits are 0. So $h_b = -1 + \frac{1}{2} = -\frac{1}{2}$, which is negative.
  therefore $h = \wedge_{j=1}^{l} h_j$ is negative since one h_j is negative.
- For $\rho(e = \{v_i, v_j, v_k\}) = (\rho(i) + \rho(j) + \rho(k), +)$
  We know that v_i, v_j, and v_k cannot all be the same color from the assumption that H is colorable. This means that $\forall j \in \chi(v)$ at least one weight in h_j will be n. This will cancel out the -1 weights so every h_j will be positive.
  therefore $h = \wedge_{j=1}^{l} h_j$ is positive since all h_j are positive.

Now we want to show the other direction.

Assume $\wedge_{j=1}^{l} h_j$ is consistent with $E(H)$.

Show H is colorable.

Assume some edge, $e = \{v_i, v_j, v_k\}$, is monochromatic. Then $\chi(v_i) = \chi(v_j) = \chi(v_k) = c$ for some color c. This means $h_c(\rho(e))$ is negative because the weights corresponding to the 1 bits of $\rho(e)$ will be -1 (i.e. $w_{ci} = w_{cj} = w_{ck} = -1$). This is a contradiction because $\rho(e)$ is defined to be positive and we have assumed $\wedge_{j=1}^{l} h_j$ is consistent with $E(H)$.

Therefore all no edges are monochromatic so H is colorable.

A method for coloring H is for vertex v, consider $\rho(v)$, color v equal to the j for the first h_j that classifies $\rho(v)$ false.

This creates a valid coloring.

We have now shown that learning an intersection of two half-spaces where the hypothesis is an intersection of k half-spaces, for a constant k, solves the consistency problem where the input is an instance of coloring a 3-uniform hyper-graph. Since coloring a 3-uniform hyper-graph is known to be NP-hard, learning an intersection of two half-spaces where the output is an intersection of k half-spaces must also be NP-hard.

■

10.4 References

[Blum, Rivest 1989] *Training a 3-Node Neural Network is NP-Complete.* Avrim Blum and Ron Rivest. Neural Networks, 5(1):117-127, 1992. Also in Advances in Neural Information Processing Systems 1 (proceedings of the 1988 NIPS conference), pp. 494-501, 1989.

[Alekhnovich, Braverman, Feldman, Klivans, Pitassi 2004] *Learnability and Automatizability.* Misha Alekhnovich, Mark Braverman, Vitaly Feldman, Adam Klivans, Toniann Pitassi. Proceedings of the 45th Foundations of Computer Science (FOCS), 2004. Invited to appear in a special issue of the Journal of Computer and System Sciencees.

[Dinur, Regev, Smyth 2002] *The hardness of 3-uniform hypergraph coloring.* In Proceedings of the 43rd Symposium on Foundations of Computer Science (FOCS), pages 33–42, 2002.