Notes on Inference Devices

Santa Fe Institute

Edward G. Huang

Summer 2018

1 Abstract

2 Introduction

2.1 Notation and Definitions

This manuscript utilizes standard notation taken from set theory and vector algebra. We clarify notation specific to Turing machine theory and inference devices.

Turing Machine Notation

$\mathbb{B}^*$	The space of all finite bit strings.
Λ	Symbol alphabet of a Turing Machine.
σ	A symbol on a Turing Machine tape.
Q	Set of finite states of a Turing Machine.
Δ	Transition function of a Turing Machine.
k	Number of tapes of a Turing Machine. The first tape is assumed to be read-only.
η	Non-halting state of a Turing Machine.

Inference Device Notation

U	Set of possible histories of the universe.
u	A history of the universe in U.
X	Setup function of an ID that maps $U \rightarrow X(U)$. A binary question concerning $\Gamma(u)$.
x	A binary question and a member of image $X(U)$.
Y	Single-valued conclusion function of an ID that maps $U \rightarrow \{-1, 1\}$. A binary answer of an ID for $X(u) = x$.
y	A single-valued answer, and member of image $Y(U) = \{0, 1\}$.
Γ	A function of the actual values of a physical variable over U, equivalent to $\Gamma(u) = S(t_i)(u)$.
γ	Possible value of a physical variable, a member of the image $\Gamma(U)$.

δ Probe of any variable V parameterized by $v \in V$ such that :

$$\delta_v(v') = \begin{cases} 1 & \text{if } v = v' \\ -1 & \text{otherwise} \end{cases}$$

$\wp$ Set of probes over $\Gamma(U)$.

$\mathcal{D} = (X, Y)$ An inference device, consisting of functions X and Y.

$\bar{F}$ Inverse. Given a function F over U, $F^{-1} = \bar{F} \equiv \{\{u : F(u) = f\} : f \in F(U)\}$.

$>$ Weak inference: a device $\mathcal{D}$ weakly infers Γ *iff* $\forall \gamma \in \Gamma(U), \exists x \in X(U)$ s.t. $\forall u \in U$, $X(u) = x \implies Y(u) = \delta_\gamma(\Gamma(u))$.

$\gg$ Strong inference: a device (X, Y) strongly infers a functions (S, T) over U *iff* $\forall \delta \in \wp(T)$ and all $s \in S(U)$, $\exists x$ *such that* $X(u) = x \implies S(u) = s, Y(u) = \delta(T(u))$.

2.2 Turing Machines

Deterministic Turing Machines

Arora and Barak denote a Turing Machine (TM) as $T = (\Lambda, Q, \Delta)$ containing:

1. An *alphabet* Λ of a finite set of symbols that T's tapes can contain. We assume that Λ contains a special blank symbol B, start symbol S, and the symbols 0 and 1.

2. A finite set Q of possible states that T's register can be in. We assume that Q contains a special start state q_s and a special halt state q_h.

3. A transition function $\Delta : Q \times \Lambda^k \to Q \times \Lambda^{k-1} \times \{L, \mathcal{S}, R\}^k$, where $k \geq 2$, describing the rules T use in performing each step. The set $\{L, \mathcal{S}, R\}$ denote the actions *Left, Stay,* and *Right*, respectively.

Suppose T is in state $q \in Q$ and $(\sigma_1, \sigma_2, \ldots, \sigma_k)$ are the symbols on the k tapes. Then $\Delta(q, (\sigma_1, \ldots, \sigma_k)) = (q', (\sigma_2', \ldots, \sigma_k'), z)$ where $z \in \{L, \mathcal{S}, R\}^k$ and at the next step the σ symbols in the last $k-1$ tapes will be replaced by the σ' symbols, the machine will be in state q, and the k heads will move *Left, Right* or *Stay.* This is illustrated in Figure 2.1.

Figure 2.2.1. The transition function Δ for a k-tape Turing Machine

$(q,(\sigma_1,\dots,\sigma_k))$				$(q',(\sigma_2',\dots,\sigma_k'),z)$				
Input symbol	Work/output symbol read	$\dots$	Current state	New work/output tape symbol	$\dots$	Move work/output tape	$\dots$	New state
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$
σ_1	σ_i	$\ddots$	q	σ_i'	$\ddots$	z_i	$\ddots$	q'
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$

Remark: Λ can be reduced to $\mathbb{B} = \{0,1\}$ and k can be reduced to 1 without loss of computational power. Then, any Turing Machine can be expressed as a partial recursive function mapping $\mathbb{B}^* \to \mathbb{B}^* \cup \eta$, where η is the undefined non-halting output. Since $|\mathbb{B}^* \times \mathbb{B}^* \cup \eta| = |\mathbb{N} \times \mathbb{N}| = |\mathbb{N}|$, the set of all Turing Machines is countably infinite.

Non-deterministic Turing Machines

Non-deterministic Turing Machines (NDTM) differ from deterministic Turing Machines by having two transition functions Δ_0, Δ_1 and a special state q_{accept}. From Arora and Barak:

> When a NDTM M computes a function, we envision that at each computational step M makes an arbitrary choice as to which of its two transition functions to apply. For every input x, we say that $M(x) = 1$ if there exists some sequence of these choices (which we call nondeterministic choices of M) that would make M reach q_{accept} on input x. Otherwise - if every sequence of choices makes M halt without reaching q_{accept} - then we say that $M(x) = 0$.

If $M(x) = 1$, we say that M accepts the input x. There are two ways to interpret the choice of update function to use in a NDTM. We can either assume that the NDTM chooses updates that will lead to an accepting state, or we can assume that the machine branches out into its choices such that it has a "computation tree" and if any of the branches reaches the accepting state then the machine accepts the input. From this second interpretation, the computational power of DTMs to NDTMs is analogous to the computational complexity of P to NP.

3 Weak Inference

Lemma *A function Γ can be weakly inferred by a device $\mathcal{D}$ if $|\Gamma^{-1}(\gamma)| \geq 2$ for any $\gamma \in \Gamma(U)$.*
Proof Let $U := \mathbb{N}$. Enumerate $\gamma \in \Gamma$ as $1, 2, \ldots, i, \ldots, n$ and define $V^i = \{u : \Gamma^{-1}(i) = u\}$. Then enumerate each element in V^i as $i_1, i_2, \ldots, i_j, \ldots, i_m$. Continue to define X and Y as

$$X(i_j) = \begin{cases} a_i & \text{if } j = 2 \\ b_i & \text{otherwise} \end{cases} \qquad Y(i_j) = \begin{cases} -1 & \text{if } j = 2 \\ 1 & \text{otherwise.} \end{cases}$$

Then for each $i \in \Gamma(U)$ choose $x = b_i$ to force $Y(X^{-1}(b_i)) = 1 = \delta_i(\Gamma(X^{-1}(b_i)) = i)$. □

Corollary *A function Γ can be weakly inferred by a device $\mathcal{D}$ if $|\Gamma(U)| \geq 3$.*
Proof Let $U := \mathbb{N}$. Enumerate $\gamma \in \Gamma(U)$ as $1, 2, \ldots, i, \ldots, n$ and define $V^i = \{u : \Gamma^{-1}(i) = u\}$. Then enumerate each element in V^i as $i_1, i_2, \ldots, i_j, \ldots, i_m$. Continue to define X and Y as

$$X(i_j) = \begin{cases} a & \text{if } i = 1 \text{ and } j = 1 \\ b & \text{if } i = 2 \text{ and } j = 1 \\ c & \text{if } i = 3 \text{ and } j = 1 \\ d & \text{otherwise.} \end{cases} \qquad Y(i_j) = \begin{cases} 1 & \text{if } i = 1 \text{ and } j = 1 \\ -1 & \text{otherwise.} \end{cases}$$

Then for each $i \in \Gamma(U)$ to force $Y(X^{-1}(x)) = \delta_i(\Gamma(X^{-1}(x)) = i)$ choose $x = a$ if $i = 1$, $x = c$ if $i = 2$, or else choose $x = b$. □

Countable Inference Theorem *A set of countable functions A^* can be weakly inferred by a device if for all $A_i \in A^*$, $\exists a \in A_i : |A_i^{-1}(a)| \geq 2$ or $|A_i(U)| \geq 3$.*

Proof Let $U := \mathbb{N}$ and fix any A^* such that for any A_i there exists an element $a \in A_i(U)$ such that $|A_i^{-1}(a)| \geq 2$. Let a_i, a_j, a_k represent any distinct three elements in $A_i(U)$. The following table represents all possible combinations of a_i, a_j, and a_k over $u \in \{1, 2, 3\}$

u	$X(u)$	$Y(u)$	$A_1(u)$	$A_2(u)$	$A_3(u)$	$A_4(u)$	$A_5(u)$	$\ldots$
1	1	-1	a_i	a_i	a_i	a_i	a_i	$\ldots$
2	2	-1	a_i	a_j	a_j	a_j	a_i	$\ldots$
3	3	1	a_i	a_k	a_j	a_i	a_j	$\ldots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$

Note that setting $X(u) = u$ and $Y(1) = -1$, $Y(2) = -1$, and $Y(3) = -1$ immediately satisfies weak inference for any function A_i whose distinct elements over $u \in \{1, 2, 3\}$ are representative of the combinations in A_1, A_2, A_3 and A_4. This holds regardless of the values of $X(u)$, $Y(u)$, and $A_{i\in\{1,2,3,4\}}(u)$ that follow for $u > 3$. This is shown by selecting x for the following cases:

Case A_1: Choose $x = 3$ for $a = a_i$. Choose $x = 2$ otherwise.
Cases A_2, A_3, A_4: Choose $x = 2$ for $a = a_i$. Choose $x = 1$ otherwise.

Now enumerate each A_i resembling $A_5 \in A^*$ as $B_4, B_5, \ldots, B_i, \ldots, B_n$. To satisfy weak inference for any B_i, define X, Y more explicitly as

$$X(u) = u, \qquad Y(u) = \begin{cases} -1 & \text{if } u = 1, 2 \text{ or if } B_i(u) = a_j, a_k : i = X(u) \\ 1 & \text{if } u = 3 \text{ or if } B_i(u) = a_i : i = X(u) \\ -1 & \text{otherwise.} \end{cases}$$

Then for each $a \in B_i(U)$, to force $Y(X^{-1}(x)) = \delta_a(B(X^{-1}(x)))$, choose $x = i$ if $a = a_i$ or otherwise choose $x = 1$. Because the choice of A^* was arbitrary, the claim holds for all A^*. $\square$

4 Strong Inference

In the next three examples we examine strong inference of integer-valued functions.

Example 2.3.1 Let $T(U) = \{0, 1\}$ and $S(U) = \{0, 1, 2\}$. We construct (X, Y) in the table at the left such that it strongly infers (S, T). The right table indicates x for each s, δ such that the definition of strong inference is satisfied:

u	$X(u)$	$Y(u)$	$S(u)$	$T(u)$
1	1	1	0	0
2	2	-1	0	0
3	3	1	1	0
4	4	-1	1	0
5	5	1	2	1
6	6	-1	2	1

$s \setminus \delta$	δ_0	δ_1
0	1	2
1	3	4
2	6	5

Example 2.3.2 Let $T(U) = \{1, 2, 3\}$ and $S(U) = \{1, 2, 3, 4, 5\}$. Again, we construct (X, Y) in the table at the left such that it strongly infers (S, T). The right table indicates x for each s, δ such that the definition of strong inference is satisfied:

u	$X(u)$	$Y(u)$	$S(u)$	$T(u)$
1	1	1	1	1
2	2	-1	1	1
3	3	1	2	1
4	4	-1	2	1
5	5	1	3	2
6	6	-1	3	2
7	7	1	4	2
8	8	-1	4	2
9	9	1	5	3
10	10	-1	5	3

$s \setminus \delta$	δ_1	δ_2	δ_3
1	1	2	2
2	3	4	4
3	6	5	6
4	8	7	8
5	10	10	9

Example 2.3.3 Let $T(U) = \{1, 2, 3\}$ and $S(U) = \{1, 2\}$. In this example, the inferred function $f : S \to T, f(s) = T(S^{-1}(s))$ is not single-valued.

u	$X(u)$	$Y(u)$	$S(u)$	$T(u)$
1	1	-1	1	1
2	2	-1	1	2
3	3	1	1	3
4	4	-1	2	1
5	5	-1	2	2

$s \setminus \delta$	δ_1	δ_2	δ_3
1	2	1	1
2	5	4	4

5 Inference of Turing Machines

Theorem *A deterministic Turing machine (Λ, Q, Δ) can be strongly inferred by a device iff*

$$\forall s \in S(U),\ |S^{-1}(s)| \geq 2.$$

This holds for both the representation of a Turing machine as a partial recursive function and the representation as an update function.

Proof First examine the partial function case. Let f be the partial recursive function that describes the given Turing machine tuple. Let $U := \mathbb{N}$. Choose any convenient single valued surjective function $S : U \to \mathbb{B}^*$ and define $T : U \to \mathbb{B}^* \cup \eta$ by $T(u) = f(S(u))$ as the single-valued function mapping U to the halting and non-halting outputs of f. Then f can be written as the single-valued mapping $S \to T$ by $f(s) = T(S^{-1}(s))$.

Enumerate the elements of $S(U)$ as $1, 2, \ldots, s, \ldots$. Let $V^s = \{u : S^{-1}(s) = u\}$ for $s \in S(U)$. Similarly enumerate the elements of V^s as $s_1, s_2, \ldots, s_{|V^s|}$. Then define X and Y as follows:

$$X(s_i) = \begin{cases} a_s & \text{if } i = 1 \\ b_s & \text{otherwise} \end{cases} \qquad Y(s_i) = \begin{cases} 1 & \text{if } i = 1 \\ -1 & \text{otherwise} \end{cases}$$

Note that the condition $|S^{-1}(s)| \geq 2$ is required to guarantee $Y(V^s) = \{1, -1\}$. For each pair $(s, \delta_{t \in T(U)})$, to force $S(u) = s$ and $Y(u) = \delta_t(T(u))$, choose $x = a_s$ if $t = T(s_1)$ or otherwise choose $x = b_s$. Since the choices of s and t were arbitrary, this holds for all (s, δ_t) pairs.

Now consider the update function that describes the given Turing machine. Recall that the update function is written as $\Delta : Q \times \Lambda^k \to Q \times \Lambda^{k-1} \times \{L, \mathcal{S}, R\}^k$, $k \geq 2$. Consider a convenient single-valued surjective function $S : U \to Q \times \Lambda^k$ representing the possible inputs for a Turing Machine and a corresponding single-valued $T : U \to Q \times \Lambda^{k-1} \times \{L, \mathcal{S}, R\}^k$ as $T(u) = \Delta(S(u))$. Observe that Δ can be written as the single-valued function $\Delta(s) = T(S^{-1}(s))$. Then define V^s, X, Y, and choose x for each (s, δ_t) as described in the preceding portion of the proof. Hence, the claim holds for the update function of a Turing machine.

To show that the condition is necessary for either representation, suppose that $|V^s| < 2$ for some s. If $V^s = \varnothing$ then there exists no x that can force $S = s$. If $|V| = 1$, then we can assign $Y(s_i) = y \in \{-1, 1\}$. However, whichever value is assigned, there exists a t such that $\delta_t(T(s_i)) \neq Y(s_i)$ since $|T(U)| \geq 2$. $\square$

Remark: Conventionally all functions over U must have a range of at least two elements. This implies that Turing machines that never halt cannot be strongly inferred.

6 Inference Complexity

Definition Let $\mathcal{D}$ be an inference device and Γ be a function over U where $X(U)$ and $\Gamma(U)$ are countable and $\mathcal{D} > \Gamma$. Let the **size** of $\gamma \in \Gamma(U)$ be written as $\mathcal{M}_{\mu:\Gamma(\gamma)} = -\ln[\int_{\Gamma^{-1}(\gamma)} d\mu(u) 1]$ such that $d\mu$ denotes a measure over U. Then the **inference complexity** of Γ with respect to $\mathcal{D}$ and measure μ is defined as:

$$\mathcal{C}_\mu(\Gamma; \mathcal{D}) \triangleq \sum_{\delta \in \wp(\Gamma)} \min_{x : X = x \implies Y = \delta(\Gamma)} [\mathcal{M}_{\mu, X}(x)]$$

Incompressibility Theorem *Let c be a positive integer. For each fixed y, every finite set A of cardinality m has at least $m(1 - m^{-c}) + 1$ elements x with $\mathcal{C}(x|y) \geq \log m - c$.*

Proof The number of programs of length less than $\log m - c$ is

$$\sum_{i=0}^{\log m - c - 1} 2^i = 2^{\log m - c} - 1.$$

Hence, there are at least $m - m2^{-c} + 1$ elements in A which have no program of length less than $\log m - c$. □

Inference Incompressibility Theorem *Let c be a positive integer. Let A^* be a countable set of finite-ranged functions and $\mathcal{D}$ be a device that infers all functions $a \in A^*$. There exists A^* and $\mathcal{D}$ such that a finite subset A of A^* with cardinality m has at least $m(1 - m^{-c}) + 1$ elements $a \in A$ such that $\mathcal{C}_\mu(a; \mathcal{D}) \geq \log m - c$. There also exists A^* and $\mathcal{D}$ such that all $a \in A$ have $\mathcal{C}_\mu(a; \mathcal{D}) = 0$.*

Proof Fix any convenient A^* and choose any countable set A from A^* with cardinality m. Construct a device $\mathcal{D}$ that can weakly infer A. This device exists by the Countable Inference Theorem. Let $U := \mathbb{B}^*$. Define $X : U \to \mathbb{N}$ as the lexicographic mapping from bit strings to integers and let $Y : U \to \{-1, 1\}$ be defined as specified in the Countable Inference Theorem.

Take $d\mu(u) = \ell(u)$ where $\ell(b)$ is the length of a bit string b. Recall that the inference complexity of $a \in A$ with respect to $\mathcal{D}$ is

$$\mathcal{C}_\mu(a; \mathcal{D}) \triangleq \sum_{\delta \in \wp(a)} \min_{x : X = x \implies Y = \delta(a)} [\mathcal{M}_{\mu,X}(x)] : \mathcal{M}_{\mu,X}(x) = -\ln \sum_{X^{-1}(x)} \ell(u)$$

By the Countable Inference Theorem, there are a class of functions that can all be weakly inferred by $u \in \{1, 2\}$. Then the minimum inference complexity for any function $a \in A$ is

$$\mathcal{C}_\mu(a; \mathcal{D}) = \sum_{\delta \in \wp(a)} \min_{x : X = x \implies Y = \delta(a)} -\ln \sum_{X^{-1}(x)} \ell(u)$$

$$= \sum_{\delta \in \wp(a)} -\ln 1 = -2 \ln 1 = 0.$$

If all $a \in A$ are instances of that class of functions then the inference complexity for all $a \in A$ would be zero. Now suppose that all $a \in A$ are in the class of functions A_5. Then the inference complexity of a is

$$\mathcal{C}_\mu(a; \mathcal{D}) = \sum_{\delta \in \wp(a)} \min_{x : X = x \implies Y = \delta(a)} -\ln \sum_{X^{-1}(x)} \ell(u)$$

$$= \sum_{\delta \in \wp(a)} \min_{x : X = x \implies Y = \delta(a)} -\ln u = -\ln 1 - \ln(u) = \ln(u).$$

Since a unique u is used to infer every a in this set, we can apply the pigeonhole principle. The number of bit strings of length less than $\log m - c$ is

$$\sum_{i=0}^{\log m - c - 1} 2^i = 2^{\log m - c} - 1.$$

Hence, there are at least $m - m2^{-c} + 1$ functions a in A which have inference complexity greater than $\log m - c$. $\square$

7 Future Work

8 Acknowledgements