Notes on Inference Devices

Santa Fe Institute

Edward G. Huang

Summer 2018

1 Notation and Definitions

Standard notation is taken from set theory and vector algebra. We clarify some notation specific to computation and inference devices.

Computation and Turing Machines

$\mathbb{B}^*$ 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 Machine.

η Non-halting state of a Turing Machine.

Inference Devices

U Set of possible histories of the universe.

u A history of the universe in U.

X Setup function of an ID that maps $U \to X(U)$. A binary question concerning $\Gamma(u)$.

x A binary question and a member of image $X(U)$.

Y Conclusion function of an ID that maps $U \to \{-1, 1\}$. 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. $\forall u \in U$, $X(u) = x \implies Y(u) = \delta_\gamma(\Gamma(u))$.

$>>$ 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 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 numbers 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 function $\Delta : Q \times \Lambda^k \to Q \times \Lambda^{k-1} \times \{L, S, R\}^k$, where $k \geq 2$, describing the rules T use in performing each step. The set $\{L, S, R\}$ 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 $z \in \{L, S, R\}^k$ 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 1.

Figure 1. The transition function Δ for a k-tape Turing Machine

$(q, (\sigma_1, \ldots, \sigma_k))$				$(q', (\sigma_2^{'}, \ldots, \sigma_k^{'}), z)$				
Input symbol	Work/output symbol read	$\ldots$	Current state	New work/output tape symbol	$\ldots$	Move work/output tape	$\ldots$	New state
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$
σ_1	σ_i	$\ddots$	q	$\sigma_i^{'}$	$\ddots$	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.

3 Inference of Turing Machines

Trivial Strong Inference Example 1

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

Trivial Strong Inference Example 2

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
6	6	1	2	3
7	7	1	3	1
8	8	1	3	2
9	9	1	3	3

Slightly Less Trivial Strong Inference Example 1

u	$X(u)$	$Y(u)$	$S(u)$	$T(u)$
1	1	1	b_1	0
2	2	1	b_2	1
3	3	1	b_3	η
4	4	1	b_4	0
5	5	1	b_5	1
6	6	1	b_6	η
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

Theorem *Every Turing Machine can be strongly inferred by an inference device.*
Proof Let $U := \mathbb{N}$. Define $S : U \to \mathbb{B}^*$ to be the lexicographic mapping of integers to binary bit strings and $T : U \to \mathbb{B}^* \times \mathbb{B}^* \cup \eta$ as a function that maps u to the Cartesian product of the space of bits strings with all possible outputs from any Turing machine.
Hence:

$$\forall \delta \in \wp(T) \text{ and all } s, \exists x \textit{ such that } X = x \implies S = s, Y = \delta(T)$$

Since the choices of S and T were arbitrary, any Turing machine can be strongly inferred by an inference device.

Theorem *Every Turing Machine can be weakly inferred by an inference device.*
Proof This follows from Proposition 4 and the preceding result.

4 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 $\mathcal{M}_{\mu:\Gamma(\gamma)} = -\ln[\int_{\Gamma^{-1}(\gamma)} d\mu(u)1]$. Then the **inference complexity** of Γ with respect to $\mathcal{D}$ and measure μ is defined as:

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

Remark: This is only a working definition and may be revised depending on its behavior.