Thermodynamic Costs of Turing Machines (**Kolchinsky˙2020**)

Daniel Briseno

February 18, 2021

Context of the Paper

Prior work on Thermodynamics of Information Processing

- Landauer cost of erasing a bit: $kT \ln 2$ (1961)
- Logically reversible computations can be performed with no heat or entropy production (1973)
- Informal argument for minimum cost of $x \mapsto y$ (1989 - 2019)
- Development of non-equilibrium statistical physics
  - Trajectory-based and stochastic thermodynamics (2013-2015)
- Thermodynamic costs of specific implementations of Turing Machines (TM)(2015-2019)

Purpose of the Paper

Thermodynamic costs of computation

- Extends results to general class of TM
- Analyzes the thermodynamic costs of $f : \mathbb{N} \nrightarrow \mathbb{N}$ on a physical implementation of a TM M
- Logical properties of f and M impose constraints on thermodynamic costs.
- Result might generalize to any implementation of a TM

Heat From Information

The Second Law of Thermodynamics

- Qualitatively, heat cannot flow from a cooler object to a hotter object.
- More Quantitatively, there exists a thermodynamic variable S, called the entropy, such that:

  $$0 \leq S_f - S_0 + \Delta Q/T$$

  where the RHS of the inequality is called Entropy Production (EP).
- An increase in "order" comes at the cost of producing Heat.

Heat From Information

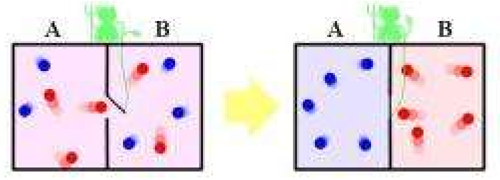


Figure: Maxwell's Demon

"... if we conceive of a being whose faculties are so sharpened that he can follow every molecule in its course, such a being, whose attributes are as essentially finite as our own, would be able to do what is impossible to us."

– James Clerk Maxwell

Heat From Information

Maxwell's Demon

- Small door can, in principle be opened with 0 EP.
- The location of each gas particle can be measured with 0 EP.
- Key lies in the demon's memory.
  - The Demon must have finite memory.
  - No matter how well-prepared, eventually the Demon must over-write (erase) one of its memory cells (**bennett˙thermodynamics˙1982**).

Heat From Information

Landauer Cost

- The Boltzmann Entropy of a single information-carrying bit is $k\ln(2)$.
- The Boltzmann Entropy of a bit that has been erased is 0.
- By the second law:

$$\begin{aligned}
0 &\le S_f - S_0 + \Delta Q/T \\
\implies 0 &\le 0 - k\ln(2) + \Delta Q/T \\
\implies kT\ln(2) &\le \Delta Q
\end{aligned}$$

and we obtain the Landauer cost.

Entropy

Measure of Disorder... Or Energy... Or Information

- Boltzmann Entropy: $S_B = k \ln(w)$
- Gibbs Entropy: $S_G = -k \sum_i P_i \ln(P_i)$
- Shannon Entropy: $\mathrm{H} = -\sum_i P_i \ln(P_i)$

Entropy

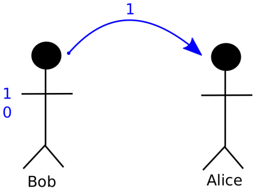


Figure: A tale as old as time.

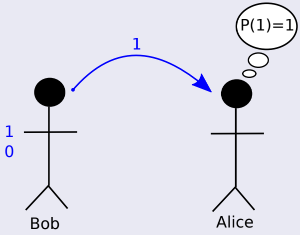
Information Content

- How much information does Bob's message give Alice?
- More quantitatively, given $\Gamma := \{0, 1\}$

$$I : \Gamma \to \mathbb{R}$$

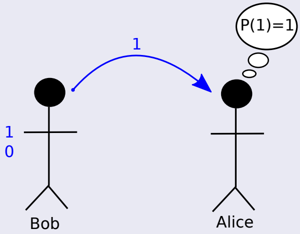
Entropy

Case 1



Entropy

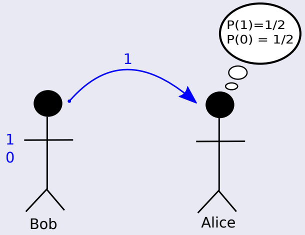
Case 1 cont.



- Bob's message provides no information.
- $I(1) = 0$

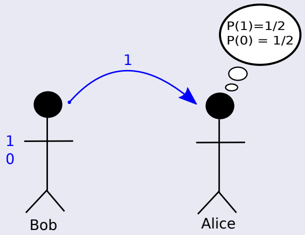
Entropy

Case 2



Entropy

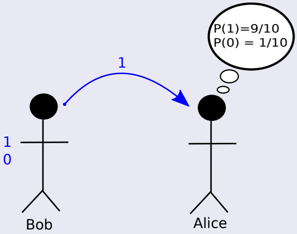
Case 2 cont.



- Bob's message provides one bit of information
- $0 < I(1) = 1$bit

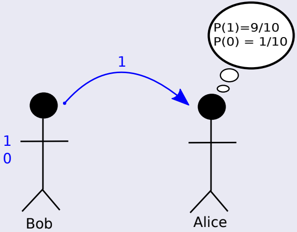
Entropy

Case3



Entropy

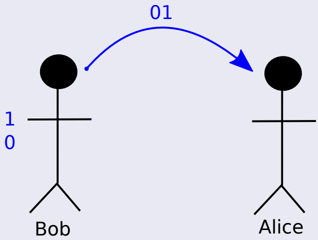
Case 3



- Bob's message removes some uncertainty
- Not as informative as in case 2
- $0 = I^{(1)}(1) < I^{(3)}(1) < I^{(2)}(1) = 1\text{bit}$

Entropy

Case 4



- Decision to send 0 independent from decision to send 1
- $I(0, 1) = I(0) + I(1)$

Entropy

Three Conditions

For any $x, y \in \Gamma$, the following three axioms must hold:

1. $P(x) = 1 \implies I(x) = 0$
2. $P(x) < P(y) \implies I(y) < I(x)$
3. $I(xy) = I(x) + I(y)$

Entropy

Information Content, or Surprisal

Single function which can satisfy these three axioms (**shannon˙mathematical˙nodate**)

$$I(x) = -\log_b(P_x)$$

Where we write $P_x = P(x)$ for notational simplicity.

- This value is called the **Surprisal**, or **Information Content**
- b sets our units of information

Entropy

The Bit: $b = 2$

- If $P(1) = P(2) = 1/2$, then $I(1) = 1$ bit.

$$\begin{aligned} I_2(1) &= -\log_2(P_1) \\ &= -\log_2\left(\frac{1}{2}\right) \\ &= 1 \end{aligned}$$

Entropy

The nat: $b = e$

- Recall the Boltzmann Entropy S_B:

$$S_B = k\ln(w)$$

- $P(w_0) = 1/w$. Thus

$$\begin{aligned} S_B &= -k\ln(P_{w_0}) \\ &= kI_e(w_0) \end{aligned}$$

Entropy

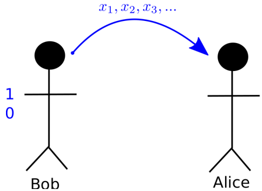
Unit Conversions

All I_b are related by the logarithm change-of-base formula:

$$I_c = \frac{1}{\log_b c} I_b$$

the factor $\frac{1}{\log_b c}$ can be thought of as a conversion factor.

Entropy



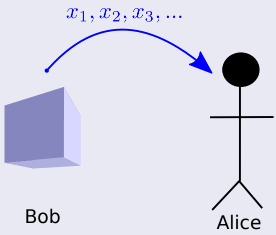
Shannon Entropy

- Single message replaced with message random variable X
- How informative is Bob to Alice?
- Expected value of Suprisal:

$$\begin{aligned} \mathrm{H}(X) &= -\sum_{i=1}^{2} P_i \log_2(P_i) \\ &= \mathbb{E}[I_2(X)] \end{aligned}$$

Entropy

Gibbs Entropy



- We replace the message alphabet Γ with a finite state-space χ

$$
\begin{aligned}
S_G &= -k \sum_{x \in \chi} P_x \ln(P_x) \\
&= k\mathbb{E}[I_e(X)]
\end{aligned}
$$

Entropy

Types of Entropies Revisited

For a system with state-space χ, $x \in \chi$, and random variable X with support χ:

- Information Content: $I_b(x) = -\log_b(P_x)$
- Boltzmann Entropy: $S_B = kI_e(w_0)$
- Shannon Entropy: $\mathrm{H} = \mathbb{E}[I_b(X)]$
- Gibbs Entropy: $S_B = k\mathbb{E}[I_e(X)] = k\,\mathrm{H}_e$

Entropy

Maxwell's Demon Revisited

- Boltzmann entropy of a information-carrying bit:

$$
\begin{aligned}
w_0 &= 1 \\
kI_e(1) &= -k\ln(P_1) = k\ln(2)
\end{aligned}
$$

- Boltzmann entropy of a bit overwritten with 0:

$$
\begin{aligned}
w_0 &= 0 \\
kI_e(0) &= 0
\end{aligned}
$$

- Second law implies:

$$
\begin{aligned}
0 &\leq 0 - k\ln(2) + \Delta Q/T \\
\implies kT\ln(2) &\leq \Delta Q
\end{aligned}
$$

CS Background

Turing Machines

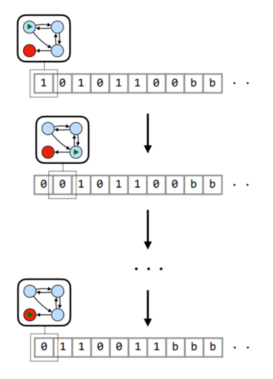


Figure: Graphical representation of a TM

CS Background

Turing Machines

Formal definition of a Turing Machine:

- A Turing machine M is a 4-tuple $M = (Q, \Sigma, q_0, \delta)$ where:
  - Q is a finite nonempty set of states.
  - Σ is a finite nonempty set of symbols.
  - $q_0 \in Q$ is the initial state of M
  - $\delta : (Q \times \Sigma) \nrightarrow (\Sigma \times \{L, R\} \times Q)$ is a partial transition function determining the symbol written on the tape, the movement of the read-write head, and the next state of the M.

CS Background

Additional Assumptions on TM M

1. $\Sigma = \{0, 1, b\}$
2. If and when M halts on an input, the tape will contain an output string $s \in \{0, 1\}^*$ followed by all blank symbols, and the pointer will be set to the start of the tape.

Assumptions do not affect the computational capabilities of M.

CS Background

Turing Machines as Partial Functions

Any computation performed by a TM M can be represented as

$$\phi_M : \{0,1\}^* \nrightarrow \{0,1\}^*$$

and $\phi_M(x) = y$ indicates that M started with input program x yields the output string y.

Universal TM

There exist Universal Turing Machines (UTM) such that given a UTM U and any TM M, there exists an interpreter program $\sigma_{U,M}$ such that

$$\phi_U(\sigma_{U,M}, x) = \phi_M(x)$$

CS Background

The Print function, an example

- There exists a FPGA (Field-programmable-gate-array) which constitutes a circuit capable of parroting back any binary string fed to it.
- This computer is also capable of printing out a binary string.

CS Background

The Print function, an example

- ϕ_D UTM which models my personal computer
- ϕ_M TM which models a "print" FPGA
- x binary string which ϕ_M can print
- $\sigma_{D,M}$ interpreter program
- $(\sigma_{D,M}, x)$ input to my UTM ϕ_D

CS Background

TM input

- The input to a TM is not generally the input to a program
- The input to a TM is the program itself, along with any necessary parameters

CS Background

Computability

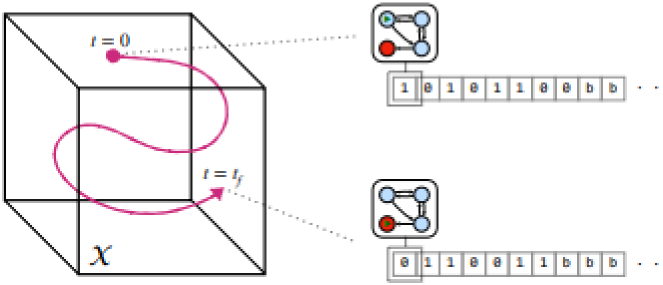
- Church Turing Thesis: A function can be calculated by a sequence of formal operations if and only if it is computable by a Turing Machine.
- Physical Church Turing Thesis: Any function implemented by a physical process can also be implemented by a Turing Machine

Realizations of a TM

Realizations and Computable Realizations

- **Physical Realization**: A physical process consistent with the laws of thermodynamics and whose dynamics correspond to the input-output map of a TM M
- **Computable Realization**: A physical realization of a TM M whose generated heat on an input program x can be determined by a computable function

Realizations of a TM



Algorithmic Information Theory

Kolmogorov Complexity

The Kolmogorov complexity K_U of a bitstring x is the length of the shortest input program that when given to a UTM U can produce x as an output:

$$K_U(x) := \min_{z:\phi_U(z)=x} \ell(z)$$

- Measure of amount of information in x

Algorithmic Information Theory

Kolmogorov Complexity of Bitstring x

$$K_U(x) := \min_{z:\phi_U(z)=x} \ell(z)$$

Kolmogorov Complexity of a Computable Function f

$$K_U(f) := \min_{M:\phi_M=f} \ell(\sigma_{U,M})$$

Conditional Kolmogorov Complexity of x Given Bitstring y

$$K_U(x|y) = \min_{z:\phi_U(z,y)=x} \ell(z)$$

Algorithmic Information Theory

Invariance Theorem

For distinct UTM U, U':

$$K_{U'}(x) = K_U(x) + O(1)$$

Thus, U is usually omitted and we write $K(x)$ for Kolmogorov complexity of x

Algorithmic Information Theory

Incompressible string x

If x is incompressible, then

$$K(x) = \ell(\text{print } x)$$

- Any program capable of producing x must contain x explicitly
- x is "maximally dense" with information

Highly compressible string π

$$K(\pi) \leq \ell\left(6\sin^{-1}\left(\frac{1}{2}\right)\right) < \ell(\text{print } \pi)$$

Algorithmic Information Theory

Input Distributions

- Input string x as random variable with probability distribution p_X
- Important example: coin flipping distribution of TM M

$$m_X^{\text{coin}}(x) := \begin{cases} 2^{-\ell(x)} & \text{if } x \in \text{dom } \phi_M \\ 0 & \text{otherwise} \end{cases}$$

- With normalizing constant $\Omega_M := \sum_{x \in \text{ dom } \phi_M} 2^{-\ell(x)}$

$$p_X^{\text{coin}}(x) = m_X^{\text{coin}}(x)/\Omega_M$$

Algorithmic Information Theory

Shannon Entropy of Distribution p_X

$$S(p_X) = -\sum_{x \in X} p_X(x) \ln p_X(x)$$

- Measure of amount of information in p_X
- $\ln \frac{1}{p_X}$: "surprisal", how unexpected, and hence informative, is x?
- $p_X(x)$: how often do we receive surprise $\ln p_X$

Algorithmic Information Theory

Entropy Production (EP)

The expected EP, written $\Sigma(p_X)$ of a physical process with initial state distribution p_X and final state distribution p_Y is:

$$\Sigma(p_X) = S(p_Y) - S(p_X) + \langle Q \rangle_{p_X} / kT$$

Thermodynamically reversible processes have $\Sigma(p_X) = 0$. EP is always nonnegative.

References