What are entropies?

A short mathematical tutorial for computer scientists

Eugene Asarin

INSTITUT DE RECHERCHE EN INFORMATIQUE FONDAMENTALE

EQINOCS workshop — Paris, 9 May 2016

Section 1

Introduction

About this tutorial

What?

- 5 or 6 classical entropy-like notions
- their classical applications
- some examples from CS

What for?

- Because knowledge is power (and fun).
- To explain the basic notion of EQINOCS project.
- To introduce basic notions for comrade speakers' talks.
- To share my vision of entropy in CS (not much, see more later)

Outline

About the entropy

Definition (almost)

Entropy of a system — a real number characterizing information content or information production of this system.

About the entropy

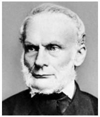
Definition (almost)

Entropy of a system — a real number characterizing information content or information production of this system.

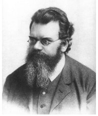
Remarks

- it was not a definition (the only precise term: "real number")
- there are multiple interesting, important and useful entropieS: in Physics, in Information theory/engineering, in Mathematics, in Computer science
- we believe it can be more interesting/useful in Computer science/engineering

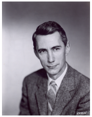
Invented by (incomplete sample)



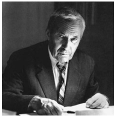
Rudolf Clausius 1822–1888



Ludwig Boltzmann 1844–1906



Claude Shannon 1916–2001



Andrey Kolmogorov 1903–1987

Yakov Sinai b. 1935

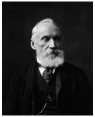
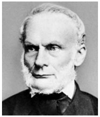
Vladimir Tikhomirov b. 1934

Roy Adler b. 1931

Rufus Bowen 1947–1978

Entropy appears in thermodynamics in 19th century

Authors

 Sadi Carnot  Lord Kelvin  Rudolf Clausius

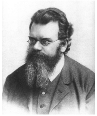
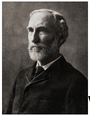
$$dS = \frac{\delta Q}{T}$$

2nd law of thermodynamics

- Planck's statement: "Every process occurring in nature proceeds in the sense in which the sum of the entropies of all bodies taking part in the process is increased"
- Corollary! No 2nd kind perpetuum motion.
- Corollary? Heat death of the Universe.

Entropy continues in statistical mechanics in 19th century

Creators of statistical mechanics


Ludwig Boltzmann

Willard Gibbs

James Maxwell

An interesting formula

$$S = k \ln W$$

with W number of microstates.

But. . .

I will not speak about entropy(-es) in physics

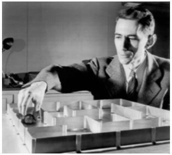
But. . .

I will not speak about entropy(-es) in physics

Only about some more mathematical ones, initiated by two giants

Two giants of 20th century

Claude Shannon



- Information theory
- Probabilistic information
- Zero-error information

Andrey Kolmogorov

- Metric entropy
- ε-entropy
- Kolmogorov complexity
- Synoptic view on entropies

Section 2

Combinatorial entropy

Towards the first definition

Question

Given a (big) finite set M, we want to describe any $x \in M$ in a file . What is the size of such file?

Towards the first definition

Question

Given a (big) finite set M, we want to describe any $x \in M$ in a file *(sequence of* 0 *and* 1). What is the size of such file?

Towards the first definition

Question

Given a (big) finite set M, we want to describe any $x \in M$ in a file *(sequence of* 0 *and* 1*)*. What is the size of such file?

Lemma

It is possible with a file of a size $\leq \log |M| + 1$. *All logs are base 2!*

Proof.

Let $m \in \{1..|M|\}$ be the position of x in M (in some, say lexicographic, order). The file F contains m in binary. $\square$

Towards the first definition

Question

Given a (big) finite set M, we want to describe any $x \in M$ in a file *(sequence of* 0 *and* 1). What is the size of such file?

Lemma

It is possible with a file of a size $\leq \log |M| + 1$.

Proof.

Let $m \in \{1..|M|\}$ be the position of x in M (in some, say lexicographic, order). The file F contains m in binary. $\square$

Lemma

For some x *(for any encoding) the file size* $> \log |M| - 2$.

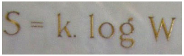
Proof.

There are only $|M|/2$ different files of size $\leq \log |M| - 2$. Hence, some $x \in M$ requires a larger file. $\square$

The first definition: combinatorial entropy

Definition (Entropy of a finite set (combinatorial))

Given a finite set M, we define its entropy by $\mathcal{H}(M) = \log |M|$.

As on Boltzmann's tomb: 

Interpretation

To specify any element of M requires $\mathcal{H}(M)$ bits of information.

- a file of size $\mathcal{H}(M)$
- or $\mathcal{H}(M)$ yes/no Q&A

The first definition: combinatorial entropy

Definition (Entropy of a finite set (combinatorial))

Given a finite set M, we define its entropy by $\mathcal{H}(M) = \log |M|$.

Interpretation

To specify any element of M requires $\mathcal{H}(M)$ bits of information.

- a file of size $\mathcal{H}(M)$
- or $\mathcal{H}(M)$ yes/no Q&A

Standard question in combinatorial entropy

Given a sequence of sets M_n explore asymptotical behavior of $\mathcal{H}(M_n)$ wrt n.

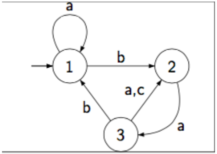
Reminder: words and languages

Terminology

- *Alphabet*: a finite set. E.g. $\Sigma = \{a, b, c\}$.
- *Letter*: its element
- *Word*: a finite sequence of letters. E.g. $w = cababac$
- *Language*: a set of words. E.g. $L = \{a, b, bbb\}$

Regular languages (an interesting subclass)

Recognized by automata, like $\longrightarrow$



We prefer *deterministic* ones: words $\leftrightarrow$ paths

$\{\varepsilon, a, b, aa, ab, ba, aaa, aab, aba, baa, bab, bac, aaaa, aaab, aaba, \dots\}$

Applying combinatorial entropy to languages

Let us do it

- Take a language $L \subset \Sigma^*$.
- Consider the words in L of length n, denote L_n
- Look at their entropies $\mathcal{H}(L_n)$
- Often $\mathcal{H}(L_n) \sim \alpha n$. This α is average per letter entropy of L.

Applying combinatorial entropy to languages

Let us do it

- Take a language $L \subset \Sigma^*$.
- Consider the words in L of length n, denote L_n
- Look at their entropies $\mathcal{H}(L_n)$
- Often $\mathcal{H}(L_n) \sim \alpha n$. This α is average per letter entropy of L.

Definition (Per letter combinatorial entropy rate of a language)

Given L its entropy is defined as

$$\mathcal{H}(L) = \limsup_{n\to\infty} \frac{\mathcal{H}(L_n)}{n}.$$

Applying combinatorial entropy to languages

Definition (Per letter combinatorial entropy rate of a language)

Given L its entropy is defined as

$$\mathcal{H}(L) = \limsup_{n\to\infty} \frac{\mathcal{H}(L_n)}{n.}$$

Two or three interpretations of $\mathcal{H}(L) = \alpha$

- Growth rate of L: i.e. $|L_n| \sim 2^{n\alpha}$
- Average information content per letter in words of L
- Take $x \in L_n$, encode it in file F, then $\alpha \approx |F|/n \Rightarrow$ α is the optimal compression rate for words in L.

Combinatorial entropy rate of languages: examples

All the words on a k-letter alphabet Σ

Entropy ?

Combinatorial entropy rate of languages: examples

All the words on a k-letter alphabet Σ

Entropy $\quad \mathcal{H}(\Sigma^*) = \limsup_{n\to\infty} \frac{\log|\Sigma^n|}{n} = \limsup_{n\to\infty} \frac{\log k^n}{n} = \log k.$

Combinatorial entropy rate of languages: examples

All the words on a k-letter alphabet Σ

Entropy $\quad \mathcal{H}(\Sigma^*) = \limsup_{n\to\infty} \frac{\log |\Sigma^n|}{n} = \limsup_{n\to\infty} \frac{\log k^n}{n} = \log k.$

All the words with 30% a, 60% b and 10% c

$$\mathcal{H}(L) = \limsup_{n\to\infty} \frac{\log \frac{n!}{(0.3n)!(0.6n)!(0.1n)!}}{n} =$$

Combinatorial entropy rate of languages: examples

All the words on a k-letter alphabet Σ

Entropy $\quad \mathcal{H}(\Sigma^*) = \limsup_{n\to\infty} \frac{\log|\Sigma^n|}{n} = \limsup_{n\to\infty} \frac{\log k^n}{n} = \log k.$

All the words with 30%a, 60%b and 10%c

$$\mathcal{H}(L) = \limsup_{n\to\infty} \frac{\log \frac{n!}{(0.3n)!(0.6n)!(0.1n)!}}{n} = \text{ (using Stirling's formula)}$$

$$\frac{\log \frac{(n/e)^n}{(0.3n/e)^{0.3n}(0.6n/e)^{0.6n}(0.1n/e)^{0.1n}}}{n} = -0.3\log 0.3 - 0.6\log 0.6 - 0.1\log 0.1 \approx 1.295$$

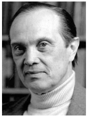
Nice formula $-\sum p_i \log p_i$, *we will see it again.*

Combinatorial entropy rate of regular languages

Pioneers

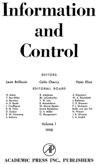
Noam Chomsky

and



George Miller

in



Problem

Given an automaton, compute the entropy rate of its language.

Combinatorial entropy rate of regular languages: solution

Computing $\mathcal{H}(L(A))$ for a deterministic A

- Remove unreachable states
- Write down the adjacency matrix M.
- Compute $\rho = \rho(M)$ - its spectral radius.
- Then $\mathcal{H} = \log \rho$.

Combinatorial entropy rate of regular languages: solution

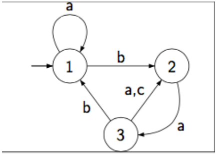
Computing $\mathcal{H}(L(A))$ for a deterministic A

- Remove unreachable states
- Write down the adjacency matrix M.
- Compute $\rho = \rho(M)$ - its spectral radius.
- Then $\mathcal{H} = \log \rho$.

Reminders

- Adjacency matrix: $M_{ij} =$ number of a such that $i \xrightarrow{a} j$.
- Spectral radius: maximal modulus of eigenvalues.

Entropy rate of regular languages — example



$$M = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 2 & 0 \end{pmatrix}$$

- Words of length 0,1,2,3,4:
  $\{\varepsilon\}; \{a, b\}; \{aa, ab, ba\}; \{aaa, aab, aba, baa, bab, bac\};$
  $\{aaaa, aaab, aaba, abaa, abab, abac, baaa, bab, baca, baba, babb\} \dots$
- Cardinalities: 1,2,3,6,11, ...
- Spectral radius: $\rho(M) \approx 1.80194$;
  entropy: $\mathcal{H} = \log \rho(M) \approx 0.84955$.

Entropy rate of regular languages — sketch of proof

Theorem

$$\mathcal{H}(L(A)) = \rho(M).$$

Proof.

- $M_{ij}^n =$ number of words w of length n such that $i \xrightarrow{w} j$
- Hence $|L_n| =$ sum of some elements of M^n
- Perron-Frobenius theory of nonnegative matrices
  $\Rightarrow |L_n| \approx \rho(M)^n \Rightarrow \mathcal{H}(L) = \log \rho(M)$

$\square$

Entropy of (regular) ω-languages — mostly the same

Reminders

- ω-*word*: an ifinite sequence of letters. E.g. *baaaaaaaaa*...
- ω-*language*: a set of ω-words, e.g. $\{271828181\ldots, 31415926\ldots\}$
- ω-*regular language*: recognized by a sort of finite automaton

Entropy of (regular) ω-languages — mostly the same

Reminders

- ω-*word*: an ifinite sequence of letters. E.g. *baaaaaaaaa* ...
- ω-*language*: a set of ω-words, e.g. $\{271828181\ldots, 31415926\ldots\}$
- ω-*regular language*: recognized by a sort of finite automaton

Definition (Staiger, entropy of an ω-language)

$$\mathcal{H}(L) = \mathcal{H}(\mathtt{pref}(L)) = \limsup_{n\to\infty} \frac{1}{n} \log |\mathtt{pref}_n(L)|$$

Entropy of (regular) ω-languages — mostly the same

Reminders

- ω-*word*: an ifinite sequence of letters. E.g. *baaaaaaaaa*...
- ω-*language*: a set of ω-words, e.g. $\{271828181\ldots, 31415926\ldots\}$
- ω-*regular language*: recognized by a sort of finite automaton

Definition (Staiger, entropy of an ω-language)

$$\mathcal{H}(L) = \mathcal{H}(\mathtt{pref}(L)) = \limsup_{n\to\infty} \frac{1}{n} \log |\mathtt{pref}_n(L)|$$

Comments, remember Ludwig's lecture @ EQINOCS

- Again: quantity of information (in bits/symbol) in words of L
- Related to Hausdorff dimension etc.

How to compute the entropy of an ω-regular language

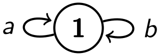
A simple algorithm

Given $L = L(A)$ an ω-regular language, where A a Büchi automaton:

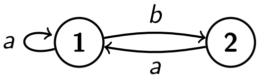
- Trim the automaton A
- Consider it as a finite automaton with all accepting states
- Determinize it.
- Write down adjacency matrix M
- Compute its maximal eigenvalue ρ.
- Return $\log \rho$.

Entropy of ω-regular languages — some examples

Example



$$\mathcal{H}(L(A)) = \log 2 = 1$$

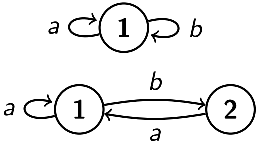


$$\mathcal{H}(L(A)) = \log \frac{1+\sqrt{5}}{2} = \log \varphi$$

- $\mathcal{H}(\Sigma^\omega) = \log |\Sigma|$;
- $\mathcal{H}(\Sigma^* b^\omega)^* = \log |\Sigma|$ (word is a prefix of L)

Entropy of ω-regular languages — some examples

Example



$$\mathcal{H}(L(A)) = \log 2 = 1$$

$$\mathcal{H}(L(A)) = \log \frac{1+\sqrt{5}}{2} = \log \varphi$$

- $\mathcal{H}(\Sigma^\omega) = \log|\Sigma|$;
- $\mathcal{H}(\Sigma^* b^\omega)^* = \log|\Sigma|$ (word is a prefix of L)

More examples and some applications in...

...Cătălin's talk this afternoon

A typical application of entropy: channel coding

Given...

- a *source*
- a *channel*

Can we transmit all?

?

A typical application of entropy: channel coding

Given...

- a *source* (possible message, contents of a file, etc.)
- a *channel* (e.g. what can be transmited by telegraph, written on a DVD, etc)

Can we transmit all?

?

Channel coding: formalizing

Given...

- a *source*: $S \subseteq \Sigma^*$
- a *channel*: $C \subseteq \Gamma^*$

(no noise, no probability in this paradigm)

Channel coding: formalizing

Given...

- a *source*: $S \subseteq \Sigma^*$
- a *channel*: $C \subseteq \Gamma^*$

(no noise, no probability in this paradigm)

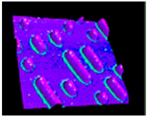
Questions

- Is it possible to transmit any source message via the channel?
- What would be the transmission speed?
- How to encode the message before and to decode it after transmission?

Writing a DVD

Description of the coding problem

- Source: $\{0,1\}^*$
- Channel: words of $\{0,1\}^*$ without blocks 11, 101, 00000000000.



Efficiency of EFMPlus

- Optimal rate for this problem: 0.5418.
- EFMPlus code used in practice, rate: $1/2$. Designed by: Kees Immink

Coding: a definition

Definition ($\phi : S \to C$, encoding with rate $\alpha \in \mathbb{Q}$)

- it is of rate α, i.e. $\alpha = \frac{|w|}{|\phi(w)|}$;
- it is injective,

Coding: a definition

Definition ($\phi : S \to C$, encoding with rate $\alpha \in \mathbb{Q}$)

- it is of rate α, i.e. $\alpha = \frac{|w|}{|\phi(w)|}$;
- it is almost injective with delay d, i.e.
  if $|w| = |w'|$ and $|u| = |u'| = d$ then
  $\phi(wu) = \phi(w'u') \Rightarrow w = w'$.

Finite state coding theorem

Information Inequality

$$\alpha\mathcal{H}(S) \le \mathcal{H}(C) \tag{II}$$

Theorem ((II) is necessary: it is easy)

If an (S, C)-encoding with rate α exists, then (II) holds.

Finite state coding theorem

Information Inequality

$$\alpha\mathcal{H}(S) \leq \mathcal{H}(C) \tag{II}$$

Theorem ((II) is necessary: it is easy)

If an (S, C)-encoding with rate α exists, then (II) holds.

Proof.

By injectivity $|S_{\alpha n}| \leq |C_{n+d}|$. Apply $\limsup \frac{1}{n} \log$ and get (II) $\square$

Finite state coding theorem

Information Inequality

$\alpha\mathcal{H}(S) \leq \mathcal{H}(C)$ (II)

Theorem ((II) is necessary: it is easy)

If an (S, C)-encoding with rate α exists, then (II) holds.

Proof.

By injectivity $|S_{\alpha n}| \leq |C_{n+d}|$. Apply $\limsup \frac{1}{n} \log$ and get (II) $\square$

Theorem ((II) is almost sufficient)

If S and C are sofic[1] *and strong (II) holds, then there exists an (S, C)-encoding realized by a finite-state transducer.*

The optimal rate...

... is $\alpha \approx \mathcal{H}(C)/\mathcal{H}(S)$

Section 3

The continuous case

The continuous case: Q&A

- Q: Given a continuous set M, how much information contains $x \in M$ (what is the file size to describe x)?

The continuous case: Q&A

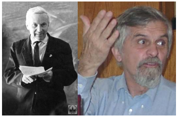
- Q: Given a continuous set M, how much information contains $x \in M$ (what is the file size to describe x)?
- A: ∞, infinitely many bits needed... it was a stupid question.

The continuous case: Q&A

- Q: Given a continuous set M, how much information contains $x \in M$ (what is the file size to describe x)?
- A: ∞, infinitely many bits needed... it was a stupid question.
- Q: Given a continuous set M, and $\varepsilon > 0$, how much information contains $x \in M$ (what is the file size to describe x with precision $\varepsilon > 0$)?

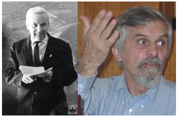
The continuous case: Q&A

- Q: Given a continuous set M, how much information contains $x \in M$ (what is the file size to describe x)?
- A: ∞, infinitely many bits needed... it was a stupid question.
- Q: Given a continuous set M, and $\varepsilon > 0$, how much information contains $x \in M$ (what is the file size to describe x with precision $\varepsilon > 0$)?
- A: Nice question, the answer by Kolmogorov & Tikhomirov is ε-entropy (and ε-capacity).



The continuous case: Q&A

- Q: Given a continuous set M, how much information contains $x \in M$ (what is the file size to describe x)?
- A: ∞, infinitely many bits needed...it was a stupid question.
- Q: Given a continuous set M, and $\varepsilon > 0$, how much information contains $x \in M$ (what is the file size to describe x with precision $\varepsilon > 0$)?



- A: Nice question, the answer by Kolmogorov & Tikhomirov is ε-entropy (and ε-capacity).
- Q: Is it relevant for computer science?
- A: IMHO yes, for timed and hybrid systems, compression of real-valued data etc.

Defining ε-entropy

Definition (ε-net)

Given M a metric space and $\varepsilon > 0$, a subset $S \subset M$ is an ε-net if $\forall x \in M \, \exists y \in S : d(x, y) < \varepsilon$

If M is compact, a finite ε-net always exists.

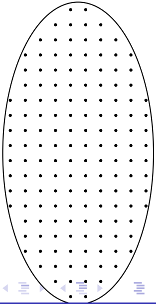
Definition (ε-entropy)

$\mathcal{H}_\varepsilon(M) = \log \min\{|S| : S \subset M \text{ an } \varepsilon\text{-net}\}$

Explanation

To describe $x \in M$ with precision ε in $\mathcal{H}_\varepsilon(M)$ bits:

- fix an optimal ε-net S;
- choose $y \in S$ such that $d(x, y) < \varepsilon$;
- write in binary the ordinal number of y in S.



Classical examples of ε-entropy and an old application

M	$\mathcal{H}_\varepsilon(M)$
A d-dimensional set of volume V	$\log(V/(2\varepsilon)^d) = O(\log(1/\varepsilon))$
1-Lipshitz functions on $[0;1]$	$O(1/\varepsilon)$
$C^k([0;1]^d)$	$O((1/\varepsilon)^{d/k})$
Analytic functions	$O(\log^2(1/\varepsilon))$

Classical examples of ε-entropy and an old application

M	$\mathcal{H}_\varepsilon(M)$
A d-dimensional set of volume V	$\log(V/(2\varepsilon)^d) = O(\log(1/\varepsilon))$
1-Lipshitz functions on $[0;1]$	$O(1/\varepsilon)$
$C^k([0;1]^d)$	$O((1/\varepsilon)^{d/k})$
Analytic functions	$O(\log^2(1/\varepsilon))$

Theorem (Vitushkin, in the context of 13th Hilbert's problem)

Exists a 1-Lipshitz f on the unit square $[0,1]^2$ which cannot be written as a term using 1-Lipshitz functions on $[0;1]$ and $+$

Classical examples of ε-entropy and an old application

M	$\mathcal{H}_\varepsilon(M)$
A d-dimensional set of volume V	$\log(V/(2\varepsilon)^d) = O(\log(1/\varepsilon))$
1-Lipshitz functions on $[0;1]$	$O(1/\varepsilon)$
$C^k([0;1]^d)$	$O((1/\varepsilon)^{d/k})$
Analytic functions	$O(\log^2(1/\varepsilon))$

Theorem (Vitushkin, in the context of 13th Hilbert's problem)

Exists a 1-Lipshitz f on the unit square $[0,1]^2$ which cannot be written as a term using 1-Lipshitz functions on $[0;1]$ and $+$

Proof.

$\mathcal{H}_\varepsilon(Lip([0,1]^2)) \approx (1/\varepsilon)^2$;

$\mathcal{H}_\varepsilon(\text{all the terms on } Lip([0,1])) \approx (1/\varepsilon)$.

Thus the former set is larger then the latter! $\square$

A new application (ongoing work)

Example ("Timed words" of duration $\leq T$)

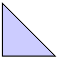
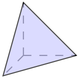
$M_T = \{t_1 a t_2 a t_3 \dots a t_k : \sum t_i \leq T\}$ How much information are there in such words (for a precision ε)?

A new application (ongoing work)

Example ("Timed words" of duration $\leq T$)

$M_T = \{t_1 a t_2 a t_3 \dots a t_k : \sum t_i \leq T\}$ How much information are there in such words (for a precision ε)?

Geometry: can we measure all that together?

$t_1 \leq T$	$t_1 + t_2 \leq T$	$t_1 + t_2 + t_3 \leq T$	$t_1 + \dots + t_k \leq T$
			k-dim simplex

A new application (ongoing work)

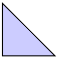
Example ("Timed words" of duration $\leq T$)

$M_T = \{t_1 a t_2 a t_3 \dots a t_k : \sum t_i \leq T\}$ How much information are there in such words (for a precision ε)?

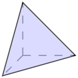
Geometry: can we measure all that together?

$t_1 \leq T$

$t_1 + t_2 \leq T$



$t_1 + t_2 + t_3 \leq T$



$t_1 + \dots + t_k \leq T$

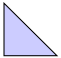
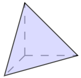
k-dim simplex

Some stores succeed

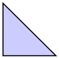
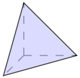
Wire at 0.9 €/m | Lino at 21€/m^2 | Sand at 75€/m^3

500 m + $40\ m^2$ + $2m^3$ =1440€

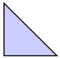
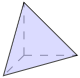
Computing ε-entropy of $M_T = \{t_1 a t_2 \dots a t_k : \sum t_i \leq T\}$

$t_1 \leq T$	$t_1 + t_2 \leq T$	$t_1 + t_2 + t_3 \leq T$	$t_1 + \cdots + t_k \leq T$
			k-dim simplex

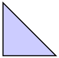
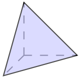
Computing ε-entropy of $M_T = \{t_1 a t_2 \dots a t_k : \sum t_i \leq T\}$

$t_1 \leq T$	$t_1 + t_2 \leq T$	$t_1 + t_2 + t_3 \leq T$	$t_1 + \dots + t_k \leq T$
			k-dim simplex
$V_1 = T$	$V_2 = \frac{T^2}{2}$	$V_3 = \frac{T^3}{6}$	$V_k = \frac{T^k}{k!}$

Computing ε-entropy of $M_T = \{t_1 a t_2 \dots a t_k : \sum t_i \leq T\}$

$t_1 \leq T$	$t_1 + t_2 \leq T$	$t_1 + t_2 + t_3 \leq T$	$t_1 + \cdots + t_k \leq T$
			k-dim simplex
$V_1 = T$	$V_2 = \frac{T^2}{2}$	$V_3 = \frac{T^3}{6}$	$V_k = \frac{T^k}{k!}$
$\lvert\varepsilon\text{-net}\rvert \approx \frac{T}{2\varepsilon}$	$\frac{T^2}{2 \cdot (2\varepsilon)^2}$	$\frac{T^3}{6 \cdot (2\varepsilon)^3}$	$\frac{T^k}{k!(2\varepsilon)^k}$

Computing ε-entropy of $M_T = \{t_1 a t_2 \dots a t_k : \sum t_i \le T\}$

$t_1 \le T$	$t_1 + t_2 \le T$	$t_1 + t_2 + t_3 \le T$	$t_1 + \cdots + t_k \le T$
			k-dim simplex
$V_1 = T$	$V_2 = \frac{T^2}{2}$	$V_3 = \frac{T^3}{6}$	$V_k = \frac{T^k}{k!}$
$\lvert\varepsilon\text{-net}\rvert \approx \frac{T}{2\varepsilon}$	$\frac{T^2}{2\cdot(2\varepsilon)^2}$	$\frac{T^3}{6\cdot(2\varepsilon)^3}$	$\frac{T^k}{k!(2\varepsilon)^k}$

Adding everything together

$$\mathcal{H}_\varepsilon(M_T) \approx \log \sum_{k=1}^{\infty} \frac{T^k}{k! 2^k \varepsilon^k} = \log(e^{T/2\varepsilon} - 1) \approx \frac{T \log e}{2\varepsilon}$$

This is the file size for any timed word in M_T up to ε.

Section 4

Shannon entropy

A simple probabilistic setting

Objects of study

- A random variable X talking values $a_1, \ldots, a_k$ with probabilities $p(a_1) = p_1, \ldots, p(a_k) = p_k$.
- A Bernoulli (iid) sequence $X_1, \ldots, X_n$ of such variables (generates a random word).

Usual question

How much information is contained in such a random word?

Shannon's solution (1948)

Definition (Shannon entropy)

$$\mathcal{H}(X) = -\sum_i p_i \log p_i.$$

Shannon's solution (1948)

Definition (Shannon entropy)

$$\mathcal{H}(X) = -\sum_i p_i \log p_i .$$

Theorem (Shannon's source coding)

A random word generated by $X_1, \dots, X_n$ can be encoded in a file of size $\approx n\mathcal{H}(X)$, with error probability $< \delta$. It is impossible with a smaller file.

Shannon's solution (1948)

Definition (Shannon entropy)

$$\mathcal{H}(X) = -\sum_i p_i \log p_i.$$

Theorem (Shannon's source coding)

A random word generated by $X_1, \ldots, X_n$ can be encoded in a file of size $\approx n\mathcal{H}(X)$, with error probability $< \delta$. It is impossible with a smaller file.

Comments

- Random words thus contain $\mathcal{H}(x)$ bits/symbol of information.
- *Déjà vu* in the combinatorial setting. Go there

Shannon's proof: the key lemma

- take $w = x_1x_2 \ldots x_n$ a random word, outcome of $X_1X_2 \ldots X_n$
- compute $s(w) = \log p(x_1) + \log p(x_2) + \cdots + \log p(x_n)$

Shannon's proof: the key lemma

- take $w = x_1 x_2 \dots x_n$ a random word, outcome of $X_1 X_2 \dots X_n$
- compute $s(w) = \log p(x_1) + \log p(x_2) + \dots + \log p(x_n)$
- by the law of big numbers with high probability $s(w) \approx \mathbb{E}s(w)$
- we have $\mathbb{E}s(w) = n\mathbb{E} \log p(X) = n \sum p_i \log p_i = -n\mathcal{H}(X)$
- with high probability $s(w) \approx -n\mathcal{H}(X)$,

Shannon's proof: the key lemma

- take $w = x_1 x_2 \dots x_n$ a random word, outcome of $X_1 X_2 \dots X_n$
- compute $s(w) = \log p(x_1) + \log p(x_2) + \cdots + \log p(x_n)$
- by the law of big numbers with high probability $s(w) \approx \mathbb{E}s(w)$
- we have $\mathbb{E}s(w) = n\mathbb{E}\log p(X) = n \sum p_i \log p_i = -n\mathcal{H}(X)$
- with high probability $s(w) \approx -n\mathcal{H}(X)$, let us exponentiate it:
- with high probability $p(w) = p(x_1) \cdot p(x_2) \cdots p(x_n) \approx 2^{-n\mathcal{H}(X)}$

Shannon's proof: the key lemma

- take $w = x_1 x_2 \dots x_n$ a random word, outcome of $X_1 X_2 \dots X_n$
- compute $s(w) = \log p(x_1) + \log p(x_2) + \dots + \log p(x_n)$
- by the law of big numbers with high probability $s(w) \approx \mathbb{E}s(w)$
- we have $\mathbb{E}s(w) = n\mathbb{E}\log p(X) = n\sum p_i \log p_i = -n\mathcal{H}(X)$
- with high probability $s(w) \approx -n\mathcal{H}(X)$, let us exponentiate it:
- with high probability $p(w) = p(x_1) \cdot p(x_2) \cdots p(x_n) \approx 2^{-n\mathcal{H}(X)}$

Lemma (AEP — almost equiprobable)

With probability $> 1 - \delta$, the probability of a random word w is close to $2^{-n\mathcal{H}(X)}$. A new interpretation of entropy

Shannon's proof continued

Definition (Typical words)

A : set of words w s.t. $2^{-n(\mathcal{H}(X)+\varepsilon)} < p(w) < 2^{-n(\mathcal{H}(X)-\varepsilon)}$.

Properties of typical words

- AEP lemma: with probability $> 1 - \delta$ a random w is in A.
- Cardinality bounds: $2^{n(\mathcal{H}(X)-\varepsilon)} < |A| < 2^{n(\mathcal{H}(X)+\varepsilon)}$.

The encoding of size $n\mathcal{H}(X)$ exists

For a random word w

- if $w \notin A$ produce an error (prob. $< \delta$)
- if $w \in A$ encode it by the ordinal number of w in lexicographic order of A

Shannon's proof finished

Reminder on the set A of typical words

- For $w \in A$ we have $2^{-n(\mathcal{H}(X)+\varepsilon)} < p(w) < 2^{-n(\mathcal{H}(X)-\varepsilon)}$.
- $P(A) > 1 - \delta$.
- $2^{n(\mathcal{H}(X)-\varepsilon)} < |A| < 2^{n(\mathcal{H}(X)+\varepsilon)}$.

$n\mathcal{H}(X)$ bits required by any encoding

Consider any encoding (with small error probability).

- Let B be the set of words we can encode with $P(B) > 1 - \delta$
- Then $P(B \cap A) > 1/2$, hence $|B| \geq |B \cap A| \geq 0.5 \cdot 2^{n(\mathcal{H}(X)-\varepsilon)}$.
- all elements of B require different files $\Rightarrow$ at least $n(\mathcal{H}(X) - \varepsilon) - 2$ bits needed.

Shannon's proof finished

Reminder on the set A of typical words

- For $w \in A$ we have $2^{-n(\mathcal{H}(X)+\varepsilon)} < p(w) < 2^{-n(\mathcal{H}(X)-\varepsilon)}$.
- $P(A) > 1 - \delta$.
- $2^{n(\mathcal{H}(X)-\varepsilon)} < |A| < 2^{n(\mathcal{H}(X)+\varepsilon)}$.

$n\mathcal{H}(X)$ bits required by any encoding

Consider any encoding (with small error probability).

- Let B be the set of words we can encode with $P(B) > 1 - \delta$
- Then $P(B \cap A) > 1/2$, hence $|B| \geq |B \cap A| \geq 0.5 \cdot 2^{n(\mathcal{H}(X)-\varepsilon)}$.
- all elements of B require different files $\Rightarrow$ at least $n(\mathcal{H}(X) - \varepsilon) - 2$ bits needed.

Shannon's entropy, what else

Nicer encoding possible

A prefix code: for each letter a take a codeword with length $\log p(a)$

Many extensions exist

- Markov chains instead of i.i.d.
- Constrained, noisy, lossy channels
- ???

I will skip one important entropy

A weakness of preceding ones

They all apply to a set M, a language L or a stochastic process X_i in order to measure information in a typical element x.

An important question

How to measure information in an x (for example a word)?

I will skip one important entropy

A weakness of preceding ones

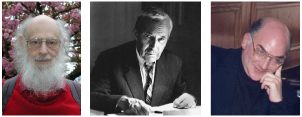
They all apply to a set M, a language L or a stochastic process X_i in order to measure information in a typical element x.

An important question

How to measure information in an x (for example a word)?

You probably know the answer

Solomonoff, Kolmogorov, Chaitin complexity



More details in Alexander Shen's talk on Wednesday.

Tired? Me too. . . A coffee break now!

Section 5

Measuring dynamical systems

What is a dynamical system?

Definition (dynamical system is a couple (X, T) with)

- X a state space
- $T : X \to X$ dynamics

Definition (trajectory)

$x, Tx, T^2x, T^3x, \dots$ (in CS this is called a run)

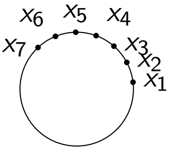
Variants and enhancements

- reversible
- continuous-time
- with some structure on X (topology, metrics, measure) and restrictions on T.

Examples of dynamical systems: "classical math"

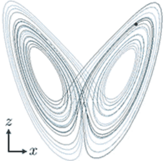
Any recurrence on $[0;1]^n$

- $Tx = \sin x$ (a stupid one)
- $T\mathbf{x} = \mathbf{x} + \mathbf{c} \mod 1$ (shift on torus)
- $T\mathbf{x} = A\mathbf{x} \mod 1$ with A integer unimodular (torus automorphism), e.g.
  $$T\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 2x+3y \\ 3x+5y \end{pmatrix} \mod 1$$



Given a differential equation $\dot{x} = f(x)$

- Tx = start from x, wait one second.
- Tx = start from x, wait until hit a plane (Poincaré map);
- continuous time. . .



Important dynamical systems: shifts

Definition (Shifts)

Fix an alphabet Σ

- State space $X = \Sigma^\omega$ or $\Sigma^\mathbb{Z}$
- Dynamics $\sigma : a_0 a_1 a_2 a_3 \cdots \mapsto a_1 a_2 a_3 a_4 \ldots$
- Probability measure on X can be added (Bernoulli, Markov)

Explanation

- State: all the future $a_0 a_1 a_2 a_3 \ldots$ (or all the eternity for bi-infinite sequences)
- Today situation : a_0
- Dynamics: (today state) $\mapsto$ (tomorrow state).

Examples of dynamical systems: computer science

Turing machine, according to Cris Moore

With moving ribbon(s) and fixed head (at 0).

- State: $Q \times \Sigma^{\mathbb{Z}}$ (control state and ribbon content)
- Dynamics T: rewrite the symbol at 0, change the state, move the ribbon, according to the program.

More general then a shift!

Subshifts $\approx$ languages

- $X \subset \Sigma^{\omega}$ or $\Sigma^{\mathbb{Z}}$, closed and shift-invariant.
- Dynamics: shift σ.
- Main example: ω-language of an automaton (w/o acceptance condition).

What about entropy?

A question about any dynamical system

At which rate does it produce information?

What about entropy?

A question about any dynamical system

At which rate does it produce information?

What for

- An interesting characteristics of systems
- Distinguishes order from chaos
- A powerful method to compare/distinguish systems.

What about entropy?

A question about any dynamical system

At which rate does it produce information?

What for

- An interesting characteristics of systems
- Distinguishes order from chaos
- A powerful method to compare/distinguish systems.

How: the general idea of symbolic dynamics

- Fix some regions
- For each trajectory consider the sequence of regions visited
- Measure entropy (combinatorial, Shannon) of such sequences

Topological dynamical systems

Definition

It is a couple (X, T)

- States: a topological space X (compact in most cases)
- Dynamics: a continuous function $T : X \to X$
- *no probability, no frills*

Towards a definition of topological entropy 1

Definition (Open cover of X and its entropy)

- A cover $\mathcal{C}$: a set of open sets with union X
- Its entropy $\mathcal{H}(\mathcal{C}) = \min\{\log|\mathcal{B}| : \mathcal{B} \subset \mathcal{C} \text{ a cover}\}$

Explanation

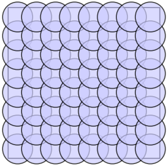
Given $x \in X$, how many bits are needed to say to which region C in $\mathcal{C}$ it belongs?

- take the smallest finite subcover $\mathcal{B}$
- take a region $B \in \mathcal{B}$ containing x
- give the ordinal number of B in $\mathcal{B}$ in binary: $\log|\mathcal{B}|$ bits.

Towards a definition of topological entropy 2

Example (Entropy of a cover)

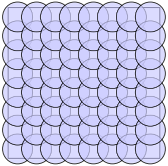
Let $X = [0,1]^2$, and $\mathcal{C}$ the cover of circles of radius < 0.1. The minimal subcover contains (I think) 64 circles, $\mathcal{H}(\mathcal{C}) = \log 64 = 6$.
Very similar to ε-entropy



Towards a definition of topological entropy 2

Example (Entropy of a cover)

Let $X = [0,1]^2$, and $\mathcal{C}$ the cover of circles of radius < 0.1. The minimal subcover contains (I think) 64 circles, $\mathcal{H}(\mathcal{C}) = \log 64 = 6$.
Very similar to ε-entropy



Definition (Join of covers)

$\mathcal{B} \vee \mathcal{C}$ consists of all $B \cap C$ with $B \in \mathcal{B}$ and $C \in \mathcal{C}$

Definition of topological entropy, Adler et al.

Definition (n-step information in system (X, T) wrt cover $\mathcal{C}$)

$$h_n(T,\mathcal{C}) = \mathcal{H}(\mathcal{C} \vee T^{-1}\mathcal{C} \vee \cdots \vee T^{-(n-1)}\mathcal{C}).$$

Explanation

We observe regions of $\mathcal{C}$ visited by a trajectory from x during n steps, and measure the (combinatorial) entropy thereof.

2

Definition of topological entropy, Adler et al.

Definition (n-step information in system (X, T) wrt cover $\mathcal{C}$)

$h_n(T, \mathcal{C}) = \mathcal{H}(\mathcal{C} \vee T^{-1}\mathcal{C} \vee \cdots \vee T^{-(n-1)}\mathcal{C})$.

Explanation

We observe regions of $\mathcal{C}$ visited by a trajectory from x during n steps, and measure the (combinatorial) entropy thereof.

Definition (Topological entropy of (X, T))

$$\mathcal{H}(T) = \sup_{\mathcal{C}} \limsup_{n\to\infty} \frac{h_n(T, \mathcal{C})}{n}$$

(information production rate)

2

Definition of topological entropy, Adler et al.

Definition (n-step information in system (X, T) wrt cover $\mathcal{C}$)

$h_n(T, \mathcal{C}) = \mathcal{H}(\mathcal{C} \vee T^{-1}\mathcal{C} \vee \cdots \vee T^{-(n-1)}\mathcal{C})$.

Explanation

We observe regions of $\mathcal{C}$ visited by a trajectory from x during n steps, and measure the (combinatorial) entropy thereof.

Definition (Topological entropy of (X, T))

$$\mathcal{H}(T) = \sup_{\mathcal{C}} \limsup_{n\to\infty} \frac{h_n(T, \mathcal{C})}{n}$$

(information production rate)

Lemma

Forget about sup*, take a generating*[2] $\mathcal{C}$

[2]

Let us compute the topological entropy for shifts

Shift $(\{a, b, c\}^\omega, \sigma\}$

- Cover $\mathcal{C}$ contains 3 opens: $a\Sigma^\omega, b\Sigma^\omega, c\Sigma^\omega$
- $\mathcal{C} \vee \sigma^{-1}\mathcal{C}$ contains 9 opens $aa\Sigma^\omega, ab\Sigma^\omega, \dots, cc\Sigma^\omega$
- In general, $\mathcal{C} \vee T^{-1}\mathcal{C} \vee \cdots \vee T^{-(n-1)}\mathcal{C}$ contains 3^n opens corresponding to n-letter prefixes.
- Cover entropies $3, 9, \dots, 3^n$ (no smaller subcovers exist)
- $\mathcal{H}(\sigma) = \frac{\log 3^n}{n} = \log 3$

Shift (Σ^ω, σ)

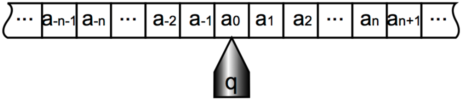
Of course $\mathcal{H}(\sigma) = \log |\Sigma|$

Topological entropy of an ω-language (subshift)

The topologic entropy of (L, σ)

- Again $\mathcal{C} \vee T^{-1}\mathcal{C} \vee \cdots \vee T^{-(n-1)}\mathcal{C}$ correspond to n-letter prefixes.
- $\mathcal{H}(L) = \limsup_{n\to\infty} \frac{\log |pref_n(L)|}{n}$ (entropy=growth rate).
- We have seen this entropy one hour ago! And we can compute it, easily.

Topological entropy of Turing machines



Theorem (Blondel&Delvenne)

Topological entropy is uncomputable for two-tape TM

Theorem (Jeandel)

Topological entropy is computable for one-tape TM!!!

It was presented at an EQINOCS meeting. . .

Remark

A TM as dynamical system is quite different from the usual perspective: all the possible tape contents should be considered!

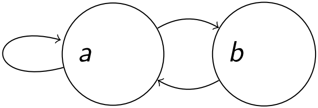
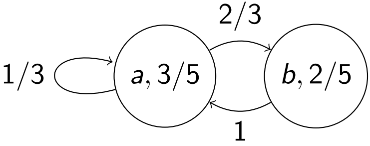
Metric dynamical systems

Definition (A metric dynamical system (X, T, μ))

- A Lebesgue[3] space X with a measure μ.
- Dynamics $T : X \to X$ (measurable)

Axiom: $\forall A : \ \mu(T^{-1}A) = \mu(A)$ (invariance of μ wrt T)

Main difference

Topological	Metric
	

[3] whatever it means

Probabilistic examples of metric dynamical systems

Bernoulli shift

$(\{a, b, c\}^{\omega}, \sigma, B)$ with Bernoulli probability B such that $p(a) = 0.3, p(b) = 0.6, p(c) = 0.1$

Probabilistic examples of metric dynamical systems

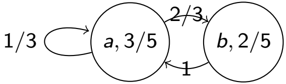
Bernoulli shift

$(\{a, b, c\}^\omega, \sigma, B)$ with Bernoulli probability B such that $p(a) = 0.3, p(b) = 0.6, p(c) = 0.1$

Markov subshift

(L, σ, M) with

- $L = (b + \varepsilon)(a^+ b)^\omega$
- M stationary Markov chain probability defined by



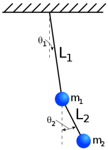
$$p(a) = 0.6, p(b) = 0.4$$

$$P = \begin{pmatrix} 1/3 & 2/3 \\ 1 & 0 \end{pmatrix}$$

Deterministic examples of metric dynamical systems

Hamiltonian systems

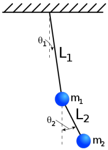
Physical systems without energy dissipation:
$\dot{p} = \partial H/\partial q;\ \ \dot{q} = -\partial H/\partial p$ with $H(p,q)$ full energy.
X : phase space; $Tx =$ position of x in 1 second;
$\mu =$ phase volume.



Deterministic examples of metric dynamical systems

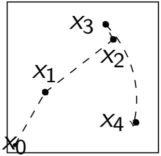
Hamiltonian systems

Physical systems without energy dissipation:
$\dot{p} = \partial H/\partial q$; $\dot{q} = -\partial H/\partial p$ with $H(p, q)$ full energy.
X : phase space; $Tx =$ position of x in 1 second;
$\mu =$ phase volume.



Torus automorphism

Like that: X: unit square; μ surface;
and $T\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 2x+3y \\ 3x+5y \end{pmatrix} \mod 1$.



Towards a definition of metric entropy

Definition (Partition of X and its entropy)

- A partition ξ: a set of disjoint sets with union X
- Its entropy $\mathcal{H}(\xi) = -\sum_{C\in\xi} \mu(C) \log \mu(C)$

Explanation

Given a μ-random $x \in X$, how many bits needed to say to which region C (of ξ) it belongs?

- As for Shannon, encode region C by a codeword with $-\log \mu(C)$ bits

Metric entropy

Definition (n-step entropy in system (X, T, μ) wrt partition ξ)

$h_n(T, \xi) = \mathcal{H}(\xi \vee T^{-1}\xi \vee \cdots \vee T^{-(n-1)}\xi)$.

Explanation

- Observe regions of ξ visited by a trajectory from x during n steps, and measure the (Shannon) entropy thereof.
- Compared to topological: open cover $\to$ partition; combinatorial $\to$ Shannon.

Metric entropy

Definition (n-step entropy in system (X, T, μ) wrt partition ξ)

$h_n(T, \xi) = \mathcal{H}(\xi \vee T^{-1}\xi \vee \cdots \vee T^{-(n-1)}\xi)$.

Explanation

- Observe regions of ξ visited by a trajectory from x during n steps, and measure the (Shannon) entropy thereof.
- Compared to topological: open cover $\to$ partition; combinatorial $\to$ Shannon.

Definition (Kolmogorov-Sinai entropy of (X, T, μ))

$$\mathcal{H}(T) = \sup_{\xi} \limsup_{n\to\infty} \frac{h_n(T, \xi)}{n}$$

Metric entropy

Definition (n-step entropy in system (X, T, μ) wrt partition ξ)

$h_n(T, \xi) = \mathcal{H}(\xi \vee T^{-1}\xi \vee \cdots \vee T^{-(n-1)}\xi)$.

Explanation

- Observe regions of ξ visited by a trajectory from x during n steps, and measure the (Shannon) entropy thereof.
- Compared to topological: open cover $\rightarrow$ partition; combinatorial $\rightarrow$ Shannon.

Definition (Kolmogorov-Sinai entropy of (X, T, μ))

$$\mathcal{H}(T) = \sup_{\xi} \limsup_{n\to\infty} \frac{h_n(T, \xi)}{n}$$

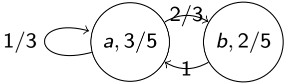
Example of computation

Bernoulli shift on $\{a, b, c\}$ with
$p(a) = 0.3, p(b) = 0.6, p(c) = 0.1$

- Partition ξ contains 3 parts: $a\Sigma^\omega, b\Sigma^\omega, c\Sigma^\omega$
- $\xi \vee \sigma^{-1}\xi$ contains 9 parts $aa\Sigma^\omega, ab\Sigma^\omega, \dots, cc\Sigma^\omega$
- In general, $\xi \vee T^{-1}\xi \vee \dots \vee T^{-(n-1)}\xi$ contains 3^n parts corresponding to n-letter prefixes.
- $h_n = -\sum_{w\in\Sigma^n} p(w)\log p(w) = \mathbb{E}_n \log p(w) = -n\mathbb{E}\log p_i = n\mathcal{H}_{\mathsf{Sh}}$
  with Shannon entropy $\mathcal{H}_{\mathsf{Sh}} = \sum p_i \log p_i \approx 1.295$
- Thus $\mathcal{H}_{\mathsf{METR}}(\sigma) = \mathcal{H}_{\mathsf{Sh}} \approx 1.295$

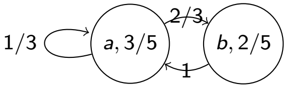
Another example

Markov subshift on $L = (b+\varepsilon)(a^+b)^\omega$



Another example

Markov subshift on $L = (b + \varepsilon)(a^+ b)^\omega$



Computing the entropy

- Partition ξ contains 2 parts: $a\Sigma^\omega, b\Sigma^\omega$
- In general, $\xi \vee T^{-1}\xi \vee \cdots \vee T^{-(n-1)}\xi$ contains parts corresponding to n-letter prefixes of L
- $\mathcal{H}(\sigma) = -\sum_i p_i \sum_j p_{ij} \log p_{ij} =$
  $-0.6\left(\frac{1}{3}\log(1/3) + \frac{2}{3}\log(2/3)\right) - 0.4 \cdot 0 \approx 0.551$

Origin of all that dynamic entropy: a problem

Definition (Isomorphism of metric dynamical systems)

(X, T, μ) and (Y, S, ν) isomorphic if exists $\varphi : X \to Y$ s.t.

- φ a bijection (upto measure 0)
- φ and φ^{-1} measurable
- φ preserves measure: $\mu(A) = \nu(\varphi(A))$
- φ compatible with dynamics: $S\varphi(x) = \varphi(Tx)$

Natural but sometimes surprising

e.g Torus automorphism isomorphic to a Markov shift!

Origin of all that dynamic entropy: a problem

Definition (Isomorphism of metric dynamical systems)

(X, T, μ) and (Y, S, ν) isomorphic if exists $\varphi : X \to Y$ s.t.

- φ a bijection (upto measure 0)
- φ and φ^{-1} measurable
- φ preserves measure: $\mu(A) = \nu(\varphi(A))$
- φ compatible with dynamics: $S\varphi(x) = \varphi(Tx)$

Natural but sometimes surprising

e.g Torus automorphism isomorphic to a Markov shift!

Problem (von Neumann, 1932 or 1941)

Are Bernoulli shifts B_2 with $p(a) = p(b) = 1/2$ and B_3 with $p(a) = p(b) = p(c) = 1/3$ isomorphic?

Classification of Bernoulli shifts: the solution

Theorem (Kolmogorov, 1957)

B_2 and B_3 are not isomorphic.

Classification of Bernoulli shifts: the solution

Theorem (Kolmogorov, 1957)

B_2 and B_3 are not isomorphic.

Proof.

$\mathcal{H}(B_2) = 1$ and $\mathcal{H}(B_3) = \log 3$, but $1 \neq \log 3$. $\square$

Classification of Bernoulli shifts: the solution

Theorem (Kolmogorov, 1957)

B_2 and B_3 are not isomorphic.

Proof.

$\mathcal{H}(B_2) = 1$ and $\mathcal{H}(B_3) = \log 3$, but $1 \neq \log 3$. □

Theorem (Ornstein, 1970)

Two Bernoulli shifts are isomorphic if and only if their entropies are equal.

Classification of Bernoulli shifts: the solution

Theorem (Kolmogorov, 1957)

B_2 and B_3 are not isomorphic.

Proof.

$\mathcal{H}(B_2) = 1$ and $\mathcal{H}(B_3) = \log 3$, but $1 \neq \log 3$. $\square$

Theorem (Ornstein, 1970)

Two Bernoulli shifts are isomorphic if and only if their entropies are equal.

And also..

Adler-Marcus analog for topological entropy.

Section 6

Everything related

Physical and others

Sorry

I don't understand physics . . .

Static case

See Sasha's talk on Shannon, combinatorial entropies and Kolmogorov complexity (on Wednesday).

Dynamic versus static 1

Naïve comparisons

1. Topological entropy is combinatorial entropy rate of region sequence
2. Metric entropy is Shannon entropy rate of region sequence
3. For languages you can tel the same story in two ways
4. There were also two communities, they get closer now!

Dynamic versus static 2

Bowen-Dinaburg's definition of topological entropy

- forget about partitions/covers
- define an n-step metrics $d_n(x, y) = \max_{i<n} d(T^i x, T^i y)$
- compute its ε-entropy $\mathcal{H}_\varepsilon(X, d_n)$
- "how much information to describe the n-step trajectory with precision ε"
- find its growth rate $\mathcal{H} = \lim_{\varepsilon\to 0} \limsup_{n\to 0} \mathcal{H}_\varepsilon(X, d_n)$

Dynamic versus static 2

Bowen-Dinaburg's definition of topological entropy

- forget about partitions/covers
- define an n-step metrics $d_n(x, y) = \max_{i<n} d(T^i x, T^i y)$
- compute its ε-entropy $\mathcal{H}_\varepsilon(X, d_n)$
- "how much information to describe the n-step trajectory with precision ε"
- find its growth rate $\mathcal{H} = \lim_{\varepsilon \to 0} \limsup_{n \to 0} \mathcal{H}_\varepsilon(X, d_n)$
- topological entropy can be phrased as ε-entropy!

Dynamic case: topological versus metric

Theorem

- $\mathcal{H}_{TOP}(X,T) = \sup_\mu \mathcal{H}_{METR}(X,T,\mu)$, *with supremum over all invariant* μ.
- *Under a weak technical condition an optimal* μ *exist:* $\mathcal{H}_{TOP}(X,T) = \mathcal{H}_{METR}(X,T,\mu)$

In other words

- Topological entropy $\geq$ the metric one
- (Often) exists an invariant measure of maximal entropy such that topological$=$ metric

$\mathcal{H}_{\mathsf{TOP}} = \max \mathcal{H}_{\mathsf{METR}}$: application

A problem and a recipe

- Given an automaton (subshift) we want to generate its words of length n equiprobably

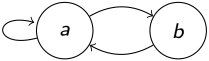
$\mathcal{H}_{\mathsf{TOP}} = \max \mathcal{H}_{\mathsf{METR}}$: application

A problem and a recipe

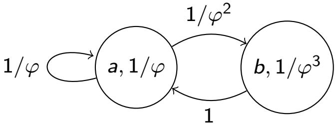
- Given an automaton (subshift) we want to generate its words of length n equiprobably
- We compute the measure of maximal entropy by simple linear algebra (Shannon-Parry measure). It is a Markov chain!
- We generate words according to this Markov chain, and it works.

$\mathcal{H}_{\mathrm{TOP}} = \max \mathcal{H}_{\mathrm{METR}}$: application continued

Example

- Aim: generate prefixes of $L = (b + \varepsilon)(a^+ b)^\omega$ (no bb)
- Automaton: 

- Topological entropy $\log \varphi$, where $\varphi =$ golden ratio.
- Maximal entropy provided by Shannon-Parry Markov chain.



- Easy to generate words!

Section 7

Summary

Summary 1

Entropy is a real number that characterizes. . .

- quantity of information (binary file size, number of binary questions, transmission time etc.)
- size or growth rate of the set of possibile beahviours
- I forgot to discuss chaos
- in fact we should distinguish between entropy (bits) and entropy rate (bits/event), but we forget to

Summary 2

We know several entropies now

- Combinatorial entropy
- Shannon entropy.
- Kolmogorov complexity (not yet)
- ϵ-entropy
- Topological entropy of dynamical systems
- Metric entropy of dynamical systems
- And they are all tightly related.

Summary 3

Entropies ($\approx$ savant cardinality arguments) are used to...

- find bounds on information transmission speed
- prove that two systems are different
- prove that they are equal

They apply to systems from computer science

- Languages and ω-languages (combinatorial and topological)
- Turing machines (topological)
- Probabilistic automata (Shannon and metric)
- anything (Kolmogorov complexity)
- more to come during next 3 days