03/08/23 - Symbolic Dynamics - Prof. Wolf

Computability on Shift Spaces

Recall: $d \geq 2$ $X \subseteq \Sigma_d^{\pm}$ $H(X) = \lim_{n\to\infty} \frac{1}{n} \log |\mathcal{L}_n(X)|$

entropy is a property of the dynamics on the space, not the space itself.
$$h_{top}(\sigma|_X) = \inf_{m\in} \frac{1}{m} \log |\mathcal{L}_m(X)|$$

Let $\Sigma_{inf} = \{X \subseteq \Sigma_d^{+/-} : X \text{ is a shift space}\}$

$\mathcal{L}(\Sigma_d^{\pm})$ is the language of the full shift, which we can give a lexicographic order $\mathcal{J} = \{w_\ell : \ell \in \mathbb{N}\}$ on the set of words.

Given $X, Y \subseteq \Sigma_{inf}$, define $d_S(X,Y) = \begin{cases} 0 & X = Y \\ \left(\frac{1}{2}\right)^\ell & \ell = \inf\ w_\ell \in \mathcal{L}(X) \cup \mathcal{L}(Y),\ w_\ell \notin \mathcal{L}(X) \cap \mathcal{L}(Y) \end{cases}$

(smallest index where the languages differ.)

Lemma: d_S is metric.

proof: 1) $d_S(X,X) = 0$ ✓ (def.)
2) $d_S(X,Y) = d_S(Y,X)$ ✓ (also def)
3) triangle inequality let $X, Y, Z \in \Sigma_{inf}$.
wts. $d_S(X,Y) \leq d_S(X,Z) + d_S(Z,Y)$ (*)
case: $X = Y$ (*) is trivial
case: $X \neq Y$. so $\exists\ \ell$ s.t. $d_S(X,Y) = (\frac{1}{2})^\ell$.
case: $w_\ell \in \mathcal{L}(X), \notin \mathcal{L}(Y)$.
$\Rightarrow w_i \in \mathcal{L}(X) \cap \mathcal{L}(Y)\ \forall i \in [1..\ell-1]$
So $d_S(X,Z) = (\frac{1}{2})^i$
case: if $i \in [1 \ldots \ell]$ then $d_S(X,Z) = (\frac{1}{2})^i \geq (\frac{1}{2})^\ell$
case: if $i > \ell$
all $w_1, \ldots, w_\ell$ in $\mathcal{L}(X)$ are also in $\mathcal{L}(Z)$
$\Rightarrow d_S(Y,Z) = (\frac{1}{2})^\ell$ which is good enough.
case: $w_\ell \in \mathcal{L}(Y), \notin \mathcal{L}(X)$
same proof.

Prop: (Σ_{inf}, d_S) is complete (every Cauchy sequence converges)

proof: let $(X_j)_{j\in\mathbb{N}}$ be a Cauchy sequence in Σ_{inf}.
This implies that $\forall n \in \mathbb{N}\ \exists J_n \in \mathbb{N}$ s.t. $\forall j \geq J$ we have $\mathcal{L}_n(X^j)$ does not depend on J.
(successive spaces must have the same words up to length n)
So we can use $\mathcal{L}_n = \mathcal{L}_n(X^J)$.
Define $L = \bigcup_n L_n$, and now we need to show L is a language

Let $w \in L$, $n = |w|$, let $v \in L_{n+1}$ arbitrary, so $v \in \mathcal{L}_{n+1}(X^j)\ \forall j \geq J_{n+1} \geq J_n$
$\Rightarrow v \in \mathcal{L}_{n+1}(X^{J_n})$, also have
$w \in \mathcal{L}_n(X^{J_n})$

Since $\mathcal{L}_n$ is a language, $\exists\ i,k \in \{0,\ldots,d-1\}$ s.t. $iw, wk \in \mathcal{L}_{n+1}$.
(extension property) But $w \in L \Rightarrow w \in L_n$ where $n = |w|$.

Now let v be a proper subword of w $\Rightarrow \exists\ k \in [1..n-1]$ s.t. $|v| = k$.
Since $J_n > J_k$, $w \in \mathcal{L}_n(X^{J_k}) \Rightarrow v \in \mathcal{L}_k(X^{J_k})$ b/c v a subword.
So $v \in L_k$.

□

then you can show $X_L = \lim_{j \to \infty} X^j$ which finishes the proof of completeness.

prop. (Σ_{inf}, d_s) is compact.

proof: relies on the fact that in a metric space compactness is equivalent to sequential compactness. (Heine - Borel)

(vibe: Cantor diagonal process w/ thinning) Note: no clear reference for this proof in the literature

Sketch: let (x_j) be a sequence in Σ_{inf}.
We will construct a convergent subsequence by induction.
(n=1) Pick $j_1 \in \mathbb{N}$ s.t. $h_1(X^{j_1}) = h_1(X^j)$ in infinately many $j \geq j_1$.

(n+1) Suppose $j_1 \cdots j_n$ are s.t. $h_1(X^{j_1}) = h_1(X^j)$ $\forall$ $j_2 \ldots j_n$ and inf. many $j \geq j_1$

$\vdots$

$h_n(X^{j_n}) = h_n(X^j)$ for infinitely many $j \geq j_n$

~~it follows that infinitely many~~ j

$\Rightarrow \exists\ j_{n+1} > j_n$ s.t.

$h_1(X^{j_1}) = h_1(X^{j_{n+1}})$

$\vdots$

$h_n(X^{j_n}) = h_n(X^{j_{n+1}})$

and $h_{n+1}(X^{j_{n+1}}) = h_{n+1}(X^j)$ for infinitely many $j \geq j_{n+1}$

So $(X^{j_n})_n$ is Cauchy $\Rightarrow$ convergent. $\blacksquare$

<u>Computability</u>

Can a computer program compute a mathematical object, within an error bound, and know when to stop.

def: let $x \in \mathbb{R}$. An <u>oracle</u> of x is a function $\varphi : \mathbb{N} \to \mathbb{Q}$ s.t. $|\varphi(n) - x| < 2^{-n}$. (the value of the number up to...)

We say $x \in \mathbb{R}$ is computable if $\exists$ a <u>turing machine</u> X which is an oracle of x.

vibes: a turing machine is any program you can write using your favorite computer program.

Remark: Rational, algebraic, and some transcendental numbers are computable. e.g. π, e,
But most real numbers are not computable, b/c programs are of finite length, so there's a countable number of them
It is hard to write down uncomputable numbers, because if I know how to explicitly write it down, I ~~then~~ have an algorithm for computing it.

Use the halting problem to get at one of these numbers.
let T_i be all turing machines.
Define $t_n = \begin{cases} 0 & \text{if } T_n \text{ halts} \\ 1 & \text{if } T_n \text{ doesn't halt} \end{cases}$

Thm: there is no universal turing machine that decides halting for any Turing machine.

03/08/23 - Symbolic Dynamics - Prof. Wolf

then let $\alpha = 0.t_1t_2t_3\ldots$ is uncomputable b/c it would be a universal turing machine

def: We say $x \in \mathbb{R}$ is upper semi-computable if $\exists$ Turing machine $\mathcal{X}: \mathbb{N} \to \mathbb{Q}$ s.t. $(\mathcal{X}(n))_n$ is non-increasing $\lim_{n\to\infty} \mathcal{X}(n) = x$

Similarly x is lower-semi computable if $-x$ is upper semi-computable.

fact: x is computable iff. x is upper and lower semi-computable

proof idea: let the upper and lower turing machines run until they are within 2^{-n} of each other.

def: let $S \subseteq \mathbb{R}^n$. We say $g: S \to \mathbb{R}$ is computable if $\exists$ turing ~~$\forall x \in S$~~, machine $\mathcal{X}$ s.t. $\forall x \in S$, oracles φ of x, $n \in \mathbb{N}$, that $\mathcal{X}(\varphi, n) \in \mathbb{Q}$ s.t. $|\mathcal{X}(\varphi, n) - g(x)| < 2^{-n}$.

It follows that computable functions are continuous
If the turing machine $\mathcal{X}$ computing $g(x)$ ~~queries~~ queries φ up to precision k then for any x' sufficiently close to x there is an oracle φ' that agrees with φ up to precision k.
$\Rightarrow |g(x) - g(x')| < 2^{-n+1}$

Are all cont. fns computable? no.
let $g: \mathbb{R} \to \mathbb{R}$. $g(x) = c$. Then g computable $\Leftrightarrow$ c computable.
This is kind of weird? e.g. does this imply $f: \mathbb{R} \to \mathbb{R}$ $f(x) = x$ not computable? no, b/c you had to be able to write down an oracle, which makes it computable.

Computable metric spaces

def: Let (X, d_X) be a complete separable (dense, countable subset) metric space.
Suppose $S = \{s_i : i \in \mathbb{N}\}$ is dense in X.
We say (X, d_X, S) is a computable metric space if $\exists$ turing machine $\mathcal{X}: \mathbb{N}^2 \times \mathbb{Z}_0 \to \mathbb{Q}$ s.t. $|\mathcal{X}(i,j,n) - d_X(s_i, s_j)| < 2^{-n}$.

(S acts like $\mathbb{Q}$, will be used for the oracle).

(Bloom-Schub-Smale model an alt. version, more algebraic, used by mathematicians)

or equivalently, $(i,j) \mapsto d_X(s_i, s_j)$ is computable
e.g. $(\mathbb{R}, |\ |, \mathbb{Q})$ is a computable metric space

def: let (X, d_X, S) be a computable metric space. let $x \in X$. We say $\varphi: \mathbb{N} \to \mathbb{N}$ is an oracle of x if $d_X|\varphi(s_n), x| < 2^{-n}$

We say $x \in X$ is computable if $\exists$ turing machine $\mathcal{X}: \mathbb{N} \to \mathbb{N}$ s.t. $\mathcal{X}$ is an oracle of x.

The notions of computable function $g: Y \to \mathbb{R}$ where $Y \subseteq X$ and also upper and lower semi-computable function generalizes accordingly.
e.g. $g: Y \to \mathbb{R}$ is computable if $\exists$ turing machine $\mathcal{X}$ s.t. $\forall x \in Y$ $\forall$ oracles φ of x, $\mathcal{X}(\varphi, n) \in \mathbb{Q}$ s.t. $|\mathcal{X}(\varphi, n) - g(x)| < 2^{-n}$

def: ~~continuity~~ computability at a point.
Let (X, d_X, S) be a computable metric space, let $Y \subseteq X$.
Let $x_0 \in X$. We say a function $g: Y \to \mathbb{R}$ is computable at x_0 if $\exists$ turing machine $\mathcal{X}$ s.t. for any oracle of x in X, $\mathcal{X}(\varphi, n)$ is $\in \mathbb{Q}$ w/ the following property:

Let $l_{\varphi,n}$ be the maximal precision that $\mathcal{X}$ queries in the calculation of $\mathcal{X}(\varphi, n)$. Then $\forall y \in S$ s.t. $\exists$ oracle φ' for y which coincides with φ up to precision $l_{\varphi,n}$ we have that $\mathcal{X}(\varphi, n) = \mathcal{X}(\varphi', n)$ and $|\mathcal{X}(\varphi, n) - g(y)| < 2^{-n}$