Lecture 1

The key for inductive proofs

- The key to an inductive proof is to fully understand the **structure** of the proofs
- The essence of inductive proof is to establish new things from the existing propositions, which occurs in various forms in Nature and Society
- Induction establishes the results one by one in **order**. So order is the key to induction.

Recursive function

For some functions f defined in the inductive step, f may not be total, in the sense that on some input x, $f(x)$ is undefined, meaning that the search enters a dead loop or unbounded searching. Those functions are called *recursively partial functions*. Usually, a function is called **recursive** if it is defined by the inductive definition above as a total function.

Turing Machine

A Turing machine is four-tuple (H, Σ, Q, Δ) of finite sets Σ, Q and Δ, satisfying:

(1) H is the location of the *reading head* of M.

(2) Σ is the alphabet representing the inputs and outputs

(3) Q is the set of states, denoted $q_0, q_1, \cdots, q_l$, with q_0 and q_l being the initial and final states, respectively.

(4) Δ is the finite set of *instructions* of the form

$$(q_i, s; q_j, s', X), \tag{10}$$

where $X = L$ or R, or M.

Turing Machine - continued

(5) Instruction $(q_i, s; q_j, s', X)$ means if the current state is q_i, and the symbol scanned by the reading head is s, then
– change the state to q_j,
– change the symbol s to s' in the cell scanned by the reading head,
– move the reading head one cell to the left if $X = L$, to the right if $X = R$, and keep unchanged if $X = M$.

Universal Turing Machine

There exists an algorithm to encode a Turing machine M to a natural number e, which generates all the Turing machines as follows:

$$M_1, M_2, \cdots, \tag{11}$$

where M_i is the Turing machine with code i, or the i-th Turing machine.

For each i, let ϕ_i be the function that is computed by M_i.

Then define U:

1. On input x, y,
2. Decode M_x,
3. Compute and output $M_x(y)$, i.e., $U(x, y) = M_x(y)$.

U is a Turing machine that simulates all M_i for all the i's., which is hence called the **universal Turing machine** - which becomes the model of the computers in the real world

Intuition of Shannon's entropy

Intuition

- $H(p)$ is the measure of the **uncertainty (information)** contained in the probability distribution p.
- The probability distribution p is **unstructured**.
- $H(p)$ is a number characterising the global uncertainty of p.
- $H(p)$ fails to support clearing of a data
- $H(p)$ is a one-dimensional metric

Question: How to define the metric of information (uncertainty) embedded in a physical system or graph that supports the analysis of the graph?

Shannon codes

Let $p = (p_1, p_2, \cdots, p_n)$ be a probability distribution of set $N = \{1, 2, \cdots, n\}$.
For each i, let $l_i = \lceil -\log_2 p_i \rceil$. We can construct a prefix-free codes α_i for i such that for each i, the length of α_i is $l(\alpha_i) = l_i$.
(**Prove this!**)

Kraft Inequality

Theorem

For any prefix code over an alphabet of size 2, the codeword lengths $l_1, l_2, \cdots, l_n$ satisfy

$$\sum_{i=1}^{n} 2^{-l_i} \leq 1. \tag{13}$$

Conversely, given a set of codeword lengths that satisfy this inequality, there exists a prefix code with these word lengths.

Information vs Computation

1. Is information useful in computation?
2. What is the role of computation in information?
   Computing is decoding information, that is, eliminating uncertainty.
3. What is the relationship between information and intelligence?
   Information is the basis of intelligence.

Lecture 2

Congruence

Definition

Given integers a, b and natural number m, we say that **a is congruent to b modulo m**, if:

$$m|(a-b).$$

In this case we write

$$a \equiv b \pmod{m}.$$

in which,
m is called *modulus* (moduli, for pl)

Remark

- $a \equiv b \pmod{m}$: a relation
- $a \bmod m$: a function, if m is fixed, and a varies.

Basic properties - II

Theorem

Let a, b be integers and m be natural number. Then:
a, b are congruent modulo m if and only if there is a k such that

$$a = b + mk.$$

$\mathbb{Z}_m$

$$\mathbb{Z}_m = \{0, 1, 2, \cdots, m-1\}.$$

Here $i \in \mathbb{Z}_m$ represents a congruence class modulo m, which is the set consists of all the numbers of the form

$$i + km$$

for all integers k.

Ring

Definition

A ring is a set R with two operations $+$ and $\cdot$, satisfying the following properties:

(1) $\langle R, + \rangle$ is a commutative group.

(2) $\langle R, \cdot \rangle$ is associative.

(3) $\langle R, +, \cdot \rangle$ is distributive.

Furthermore, if $\langle R, \cdot \rangle$ is commutative, we say that $\langle R, +, \cdot \rangle$ is a commutative ring.

Representations

- Decimal: base 10
- Binary: base 2
- Octal: base 8
- Hexadecimal: base 16

Why not base 1?

Theorem

Let b be a natural number greater than 1*. Then, for every positive integer n, there exists a unique base b representation of n, that is, there is a unique* $k+1$*-tuple* $(a_0, a_1, \cdots, a_k)$ *satisfying:*

1. *for each j,* $0 \leq a_j < b$*, and* $a_k \neq 0$*,*
2. $$n = a_k b^k + a_{k-1} b^{k-1} + \cdots + a_1 b + a_0.$$

The fundamental theorem of arithmetic

Definition

We say that a natural number n is *prime*, if there are no $a, b < n$ such that $n = a \cdot b$, and *composite*, otherwise.

Intuition: Primes are the “atomics" or “building blocks" of numbers.

Theorem

Every integer greater than 1 *can be uniquely represented by the following form*

$$n = p_1 p_2 \cdots p_k,$$

where p_j*'s are primes in increasing order.*

Basic property

Theorem

If n is composite, then there exists a prime $p \leq \sqrt{n}$ such that $p|n$.

Towards a contradiction. If $n = q_1 q_2 \cdots q_l$, for $l \geq 2$ and for primes q_j. Then each $q_j > \sqrt{n}$, implying $q_1 q_2 > n$. A contradiction.

Significance: Anyway, the theorem reduces somehow the search space for a prime factor of a natural number, leading to some algorithms.

The Prime Number Theorem

For every natural number x, let N_x be the number of primes less than or equal to x.

Theorem
(1896)

$$\lim_{x\to\infty} \frac{N_x}{\frac{x}{\ln x}} = 1. \tag{14}$$

Graph as An Extension of Relations

We know that

(i) Relation is an extension of functions, and

(ii) Graphs are extensions of relations.

Questions

1) Why?

2) Are graphs well-defined mathematical model?

Elements of a Graph

How to understand a graph? A graph contains

- **Syntax**
- **Semantics**
- **Structural functions**

Volume

Definition

Let $G = (V, E)$ be an undirected graph and $S \subseteq V$. The *volume* of S in G is defined by

$$\mathrm{vol}(S) = \sum_{u \in S} d(u). \tag{15}$$

Wheel W_n

For $n > 3$, the wheel W_n consists of a cycle C_{n-1} and a center vertex u such that for each vertex v of cycle C_{n-1}, there is an edge between u and v.

Primal and Dual

Theorem

If an LP has an optimal solution, so does its dual, and at optimality their costs are equal.

Lecture 3

Tautology, 永真式 (Valid)

A proposition ϕ is called a **tautology**, or called **valid**, if any any assignment of the atomic Beoolean variables, ϕ takes the value true.

Duality

Given a Boolean expression E, the dual expression of E is obtained from E by

- interchange $+$ and $\cdot$
- interchange 0 and 1

Sum of Productions

Definition

A literal is a Boolean variable or its complement. A minterm is the product of some literals.

Theorem

Every Boolean function can be expressed by a sum of productions.

This is similar to the case of Boolean expressions: Every Boolean expression can be expressed as a disjunctive normal form. (DNF)

Product of Sums

Theorem

Every Boolean function can be expressed as the product of sums of literals.

This is similar to: Every expression can be expressed as a conjunctive normal form (CNF).

Boolean Circuits as Model of Computation

A Boolean circuit is a diagram showing how to derive an output from a binary input string by applying a sequence of basic Boolean operations OR ($\vee$), AND ($\wedge$) and NOT ($\neg$) on the input bits.

Definition

For every $n \in \mathbb{N}$, and n-input, single-output Boolean circuit is a directed acyclic graph C (DAG) with n sources (input vertices) and one sink (output vertex). All nonsource vertex are called gates and are labelled with one of $\vee$, $\wedge$ and $\neg$. The vertices labelled $\vee$ and $\wedge$ have fin-in 2, and the vertices labelled with $\neg$ have fin-in 1. The size of C, denoted by $|C|$, is the number of vertices in C.

Circuit Families

Definition

Let $T : \mathbb{N} \to \mathbb{N}$ be a function. A $T(n)$-size circuit family, is a sequence $\{C_n\}_{n\in\mathbb{N}}$ of Boolean circuits, where C_n has n inputs and single output, and have size $|C_n| \leq T(n)$, for each n.
We say that a language L is in $\mathrm{SIZE}(T(n))$, if there is a $T(n)$-size circuit family $\{C_n\}$ such that for every n and every $x \in \{0,1\}^n$,

$$x \in L \iff C_n(x) = 1.$$

Key: Nonuniform computation

The Class NC

Definition

For every d, a language L is in NC^d if L can be decided by a family of circuits $\{C_n\}$, where C_n has $\mathrm{poly}(n)$ size and depth $O(\log^d n)$.

The class NC is:

$$\mathrm{NC} = \cup_{i \geq 1} \mathrm{NC}^i.$$

We define uniform NC to be the circuits that is generated by a log space algorithm.

The Class AC

Definition

The class AC^i is defined similarly to NC^i except that gates are allowed to have unbounded fan-in.
The class AC is:

$$\mathrm{AC} = \cup_{i \geq 0} \mathrm{AC}^i.$$

Nondeterministic Turing Machine

NP: nondeterministic polynomial time

A **nondeterministic Turing machine** M has two **transition functions** δ_0 and δ_1 such that

$$\delta_0(q, s) = (q_0, s_0, X_0)$$

$$\delta_1(q, s) = (q_1, s_1, X_1)$$

When the the state is q reading s, there are two choices either $\delta_0(q, s)$ or $\delta_1(q, s)$ to proceed to the next step of the computation.

For every input x, $M(x) = 1$ if and only if there exists a sequence of choices $\sigma_1, \sigma_2, \cdots, \sigma_l$ make M halts with an accepting state q_{accept}. Otherwise, then $M(x) = 0$.

We say that M runs in $T(n)$, if for any choices of the transition functions, M halts with $T(|x|)$ steps.

Configuration Graph

A **configuration** of a Turing machine M is a tuple consists of the **state**, the **position of the head**, the **symbol** that is scanned by the head, and the **sequence of nonblank cells** of the work tape.

A **configuration graph** of a Turing machine on input x is the graph of the configurations of the computations on input x following the directions instructed by the instructions of the Turing machine.

Space Complexity Classes

$$
\begin{aligned}
\mathrm{PSPACE} &= \cup_{c>0}\mathrm{SPACE}(n^c) \\
\mathrm{NPSPACE} &= \cup_{c>0}\mathrm{NSPACE}(n^c) \\
\mathrm{L} &= \mathrm{SPACE}(\log n) \\
\mathrm{NL} &= \mathrm{NSPACE}(\log n)
\end{aligned}
$$

The Complexity Classes

$$\mathrm{L} \subseteq \mathrm{NL} \subseteq \mathrm{P} \subseteq \mathrm{NP} \subseteq \mathrm{PSPACE} \tag{4}$$

Theorem
(Hartmanis, 1965)

$$\mathrm{L} \subsetneq \mathrm{PSPACE}.$$

Lecture 4

The Markov Inequality

Let X be a discrete random variable and $f(x)$ be any real-valued function. The *expectation* of $f(X)$ is defined by

$$E[f(X)] = \sum_x f(x) \cdot \Pr[X = x]. \tag{26}$$

Theorem

(Markov Inequality) Let Y be a random variable assuming only non-negative values. Then for all $t \in \mathbb{R}^+$,

$$\Pr[Y \geq t] \leq \frac{E[Y]}{t}. \tag{27}$$

Equivalently,

$$\Pr[Y \geq k \cdot E[Y]] \leq \frac{1}{k}. \tag{28}$$

Chebyshev's Inequality

Theorem

(Chebyshev's Inequality) Let X be a random variable with expectation μ and standard deviation σ. Then for any $t \in \mathbb{R}^+$,

$$\Pr[|X - \mu| \geq t \cdot \sigma] \leq \frac{1}{t^2}. \tag{30}$$

The kth central moment

Definition

For $k \in \mathbb{N}$, the kth *moment* and the kth *central moment* of a random variable X are defined by

$$\mu_X^k = E[X^k]$$

$$\sigma_X^k = E[(X - E[X])^k].$$

The expected value is the first moment, the variance is the 2nd central moment.

L_p-norm

L_p-norm of v, $p \geq 1$,

$$||v||_p = (\sum_{i=1}^{n} |v_i|^p)^{1/p}$$

$p = 2$, L_2-norm, the Euclidean norm

$$||v||_2 = (\sum_{i=1}^{n} |v_i|^2)^{1/2}$$

$p = 1$, L_1-norm

$$||v||_1 = \sum_{i=1}^{n} |v_i|$$

$p = \infty$, L_∞-norm

$$||v||_\infty = \max_i |v_i|.$$

Lecture 5

Matrices product

Let $\mathbb{F}$ be a finite field, $\mathbb{Z}_p$ for some prime, p say. Let A, B and C be $n \times n$ matrices over $\mathbb{F}$.
To test whether or not $AB = C$, naive approach is to compute the matrix product and compare - in time complexity $O(n^3)$.
By fingerprinting, we test as follows:
Tester $\mathcal{T}$:

(1) Let r be a vector chosen randomly and uniformly from $\{0,1\}^n$ (of course could be any other field, $\mathbb{F}^n$ say)

(2) Let $x = Br$, $y = Ax$ and $z = Cr$.
(Time complexity $O(n^2)$.)

(3) If $y = z$, then accepts, and rejects, otherwise.

Proof

If $AB = C$, then $\mathcal{T}$ accepts with probability 1.
Suppose that $AB \neq C$. Let $D = AB - C = (d_{ij})$. Suppose $d_{11} \neq 0$. For the random vector

$$r = \begin{pmatrix} r_1 \\ r_2 \\ \vdots \\ r_n \end{pmatrix} \tag{1}$$

If $Dr = 0$, then $d_{11}r_1 + d_{12}r_2 + \cdots + d_{1n}r_n = 0$, giving

$$r_1 = -\frac{d_{12}r_2 + \cdots + d_{1n}r_n}{d_{11}}, \tag{2}$$

which occurs with probability at most $\frac{1}{2}$.
Therefore, the probability that $\mathcal{C}$ accepts is at most $\frac{1}{2}$.

Polynomial identity test

Theorem

Let $\mathbb{F}$ be a finite filed, and $Q(x_1, \cdots, x_n) \in \mathbb{F}[x_1, \cdots, x_n]$ be a multivariate polynomial of total degree d over $\mathbb{F}$. Let $S \subset \mathbb{F}$, and let $r_1, \cdots, r_n$ be chosen independently and uniformly at random from S. Then,

$$\Pr[Q(r_1, \cdots, r_n) = 0 \mid Q(x_1, \cdots, x_n \not\equiv 0] \leq \frac{d}{|S|}. \tag{3}$$

Proof

If $Q(x_1, \cdots, x_n) \equiv 0$, then the probability that $Q(r_1, \cdots, r_n) = 0$ is 1.
Suppose that $Q \not\equiv 0$.
By induction on n. $n = 1$, done before. Suppose the theorem holds for all $n' < n$ and $n > 1$.
Let

$$Q(x_1, x_2, \cdots, x_n) = \sum_{i=0}^{k} x_1^i Q_i(x_2, \cdots, x_n), \tag{4}$$

for $k > 0$.
By the choice of k, the coefficient $Q_k(x_2, \cdots, x_n)$ of x_1^k is not identically zero, and the total degree of Q_k is $d - k$.

Proof - continued

By inductive hypothesis,

$$\Pr[Q_k(r_2, \cdots, r_n) = 0] \leq \frac{d-k}{|S|}. \tag{5}$$

Assume $Q_k(r_2, \cdots, r_n) \neq 0$. Let

$$q(x_1) = \sum_{i=0}^{k} x_1^i Q_i(r_2, \cdots, r_n).$$

Then

$$\Pr[q(r_1) = 0] \leq \frac{k}{|S|}. \tag{6}$$

Therefore,

$$\Pr[Q(r_1, r_2, \cdots, r_n) = 0] \leq \frac{d-k}{|S|} + \frac{k}{|S|} = \frac{d}{|S|}.$$

Identity of data

Alice and Bob share the data D initially. During the procedure of processing, the data may be corrupted. So they want to make sure that their data A and B are are same.
However, the data A and B are huge, for which verification of equality is not easy.
By fingerprinting, we may check easily as follows:

1. To transform A and B to $a = (a_1, \cdots, a_n)$ and $b = (b_1, \cdots, b_n)$ of numbers in a universe $\mathbb{F}^n$.
2. For a prime p, define the fingerprint by

$$f_p(x) = x \bmod p. \tag{7}$$

3. Randomly pick a prime p,
if $f_p(a) = f_p(b)$, then accept, and reject, otherwise.

General ideas of fingerprinting

- Characterise the two objects as polynomials A and B
- Randomly and uniformly choose a random number r in $\mathbb{Z}_p$, written $r \in_{\mathrm{R}} \mathbb{Z}_p$.
- The fingerprints is $A(r)$ and $B(r)$ for random r, in a field $\mathbb{Z}_p$ for some prime p
- If $A \equiv B$, then accepts with probability 1, and if $A \not\equiv B$, the probability of acceptance is at most $\frac{k}{p}$.
- The n-bit comparison is reduced to compare only $O(\log n)$ bits.

Hash Table

(i) It is a table T of n cells, indexed by

$$N = \{0, 1, \cdots, n-1\}.$$

(ii) A hash function is a function of the form:

$$h: \ M \to N,$$

where $M = \{0, 1, \cdots, m-1\}$ and $m >> n$.

(iii) Each cell in table T allows to encode an element of M, i.e., with size $\log m$.

(iv) The hash function is a fingerprint function for the keys in a large set M to the small set N of fingerprints (cells)

(v) Fingerprint function h ensures that for distinct keys $x \neq y$, the probability that the cells $h(x)$ equals $h(y)$, i.e., $h(x) = h(y)$, is small, so that collisions occur with a only small probability

The construction of hash functions

Fix m and n. Choose a prime $p \geq m$. We will work over the field $\mathbb{Z}_p$.

1. Let $g : \mathbb{Z}_p \to \mathbb{N}$ be the function

$$g(x) = x \bmod n, \tag{10}$$

   for some small number n, - the length of the hash table.

2. Define

$$f_{a,b}(x) = ax + b \bmod p. \tag{11}$$

$$h_{a,b}(x) = g(f_{a,b}(x)). \tag{12}$$

3. Let $H = \{h_{a,b} \mid a, b \in \mathbb{Z}_p, a \neq 0\}$. Then H is a family of hash functions.

RSA

Suppose that $n = pq$ for some primes p, q and $p \neq q$.
Suppose that d and e are numbers satisfying:

$$de = 1 + k(p-1)(q-1), \tag{22}$$

for some integer k.
Then the encode is

$$E : M \to C = M^e \bmod n. \tag{23}$$

Public key

- n can be public
- one of the e and d can be public
- Both p, q are kept for privacy.
- One of e and d is kept for privacy.

The Assumption

1) Finding one of d (or e) from the public e (or d) is hard, without given p and q,

2) Finding the prime factors p, q for n is hard.

Key Agreement Protocol

(1) Alice and Bob agreed a prime p and its primitive root a.

(2) Alice chooses a secret number k_1 and sends $a^{k_1} \bmod p$ to Bob.

(3) Bob chooses his own key k_2 and sends a^{k_2} to Alice

(4) Alice computes

$$(a^{k_2})^{k_1} \equiv a^{k_1 k_2} \bmod p.$$

(5) Bob computes

$$(a^{k_1})^{k_2} \equiv a^{k_2 k_1} \bmod p.$$

(6) Alice and Bob Achieved their shared key:

$$a^{k_1 k_2} \bmod p.$$

Lecture 7

Number of Bits Required to Guess the Item

Therefore, the Shannon entropy of a probability distribution can be approximately understood as the minimum average length of the binary representation of the item chosen by probability distribution p.
This gives an intuitive understanding of the Shannon entropy.
In addition, the Shannon entropy can be regarded as the minimum number of guesses to determine the item chosen by the probability distribution p.
However, **probability distribution is structure free**.

Encoding Tree

Definition

(Encoding tree of graphs) Let $G = (V, E)$ be an undirected and connected network. We define the *encoding tree T of G* as a tree T with the following properties:

(1) For the root node denoted λ, we define the set $T_\lambda = V$.

(2) For every node $\alpha \in T$, the immediate successors of α are $\alpha\hat{}\langle j\rangle$ for j from 1 to a natural number N ordered from left to right as j increases, where every internal node has at least two immediate successors. Therefore, $\alpha\hat{}\langle i\rangle$ is to the left of $\alpha\hat{}\langle j\rangle$ written as $\alpha\hat{}\langle i\rangle <_{\mathrm{L}} \alpha\hat{}\langle j\rangle$, if and only if $i < j$.

(3) For every $\alpha \in T$, there is a subset $T_\alpha \subset V$ that is associated with α.
For α and β, we use $\alpha \subset \beta$ to denote that α is an initial segment of β. For every node $\alpha \neq \lambda$, we use α^- to denote the longest initial segment of α, or the longest β such that $\beta \subset \alpha$.

Encoding Tree - II

(4) For every i, $\{T_\alpha \mid h(\alpha) = i\}$ is a partition of V, where $h(\alpha)$ is the height of α (note that the height of the root node λ is 0, and for every node $\alpha \neq \lambda$, $h(\alpha) = h(\alpha^-) + 1$).

(5) For every α, T_α is the union of T_β for all β's such that $\beta^- = \alpha$; thus, $T_\alpha = \cup_{\beta^- = \alpha} T_\beta$.

(6) For every leaf node α of T, T_α is a singleton; thus, T_α contains a single node of V.

Encoding Tree - III

(7) For every node $\alpha \in T$, if $T_\alpha = X$ for a set of vertices X, then we say that α is the **codeword** of X, and that X is the **marker** of α.

(8) For every vertex $v \in V$, there is a leaf node $\alpha \in T$ such that $T_\alpha = \{v\}$, that is, there is a unique **codeword** of v in T.

(9) Every leaf node in T is a codeword of a unique vertex (**marker**) in V.

Therefore, the set of the leaf nodes in T is the set of codewords of all the vertices in G.

Structural Entropy by an Encoding Tree

Definition

(Structural entropy of a graph by an encoding tree) For an undirected and connected network $G = (V, E)$, suppose that T is an encoding tree of G.

We define the **structural entropy of G by the encoding tree T** as follows:

$$\mathcal{H}^T(G) = -\sum_{\alpha \in T, \alpha \neq \lambda} \frac{g_\alpha}{2m} \log_2 \frac{V_\alpha}{V_{\alpha^-}}, \tag{4}$$

where g_α is the number of edges from nodes in T_α to nodes outside T_α, V_β is the volume of set T_β, namely, the sum of the degrees of all the nodes in T_β.

Structural Entropy

Definition

(Structural entropy) Let $G = (V, E)$ be a connected network.

- We define the **structural entropy of** G as follows:

$$\mathcal{H}(G) = \min_{T}\{\mathcal{H}^{T}(G)\}, \tag{6}$$

where T ranges over all of the encoding trees of G.

One-dimensional Structural Entropy

Let $G = (V, E)$ be a connected graph, and p be the degree distribution of G. Then

$$\mathcal{H}^1(G) = H(p), \tag{7}$$

written $H(G)$, called **Shannon entropy of** G.

Intuition

Given a connected graph G, the structural entropy $\mathcal{H}(G)$ is the minimum number of bits required to determine the codeword of an encoding tree of the graph for the vertex that is accessible from random walk with stationary distribution in the graph.

Compressing Information by Encoding Tree

Definition

(Compressing information of a graph by an encoding tree) For an undirected and connected network $G = (V, E)$, suppose that T is an encoding tree of G.

We define the **compressing information of G by the encoding tree T** as follows:

$$\mathcal{C}^T(G) = -\sum_{\alpha \in T, \alpha \neq \lambda} \frac{V_\alpha - g_\alpha}{2m} \log_2 \frac{V_\alpha}{V_{\alpha^-}}, \tag{20}$$

where g_α is the number of edges from nodes in T_α to nodes outside T_α, V_β is the volume of set T_β, namely, the sum of the degrees of all the nodes in T_β.

Decoding Information by Encoding Tree

Definition

(Decoding information of a graph by an encoding tree) For an undirected and connected network $G = (V, E)$, suppose that T is an encoding tree of G.

We define the **decoding information of G by the encoding tree T** as follows:

$$\mathcal{D}^T(G) = \mathcal{H}^1(G) - \mathcal{H}^T(G). \tag{21}$$

Compressible Graphs

Definition

Given G, k and ρ, we say that G is (n, k, ρ)-compressible, if:

$$\rho^k(G) \geq \rho,$$

where

$$\rho^k(G) = \frac{\mathcal{C}^k(G)}{\mathcal{H}^1(G)}$$

Compressing and Decoding Principle

Theorem

Let $G = (V, E)$ be a connected network. Then:

$$\mathcal{C}(G) = \mathcal{H}^1(G) - \mathcal{H}(G) = \mathcal{D}(G). \tag{26}$$

Therefore, **any information lost in the compression of data can be losslessly decoded by an encoding tree, the decoder**.

This means that

For either unstructured or structured data, data compression will never loss any information

Principles of Structural Information Theory

- **Structural entropy minimisation** is the principle for data analysis
- **Encoding tree** optimizes data structure of massive data
- **Encoding** eliminates the uncertainty that is embedded in a complex system
- **Structural information decoding** is a general model for information processing
- **Information is generated by structures**, providing a new understanding of the notion of information

Structural Dimension

Definition

Let $G = (V, E)$ be an undirected and connected graph.

1. We define the structural dimension of G to be the least k such that

$$\mathcal{H}^k(G) = \mathcal{H}(G)$$

   We use $D(G)$ to denote the structural dimensional of G.

2. We define the upper structural dimension of G to be the greatest k such that

$$\mathcal{H}^k(G) = \mathcal{H}(G)$$

   We use $D^U(G)$ to the upper structural dimension of G.

Algebraic Understanding of the Structural Entropy

We have shown that for any undirected and connected graph G, if the two-dimensional structural entropy $\mathcal{H}^2(G)$ is small, then there is a large k such that the k-th largest eigenvalue of the Laplacian of G is small (less than ϵ).
This suggests that the structural entropy of graphs is closely related to the distribution of the eigenvalues of the Laplacian of graphs, leading to a new direction for graph theory.

Game vs Information

Information must be the basis of game. We have shown that structure and randomness are key to the generation of information. The fundamental questions are hence:

1. What are the roles of structure and randomness in game?
2. What is the role of information in game
3. Is there an information theoretical direction of game theory?