Course «Information theory». Very brief lecture notes.

03.12.2019. Lecture 11.

1. Probabilistic model of a noisy channel. A *discrete memoryless channel* is formally defined by two alphabets (the alphabet of *input signals* A and the alphabet of *output signals* B) and by the matrix of signals transformation probabilities (q_{ij}) (for $i = 1, \ldots, |A|$ and $j = 1 \ldots, |B|$), where

$$q_{ij} = \text{Prob}[\text{ the outcome of the channel is } b_j \mid \text{input of the channel is } a_i \text{]}. \quad (*)$$

We assume that every unite of time (e.g., every millisecond) the channel gets a symbol a_i on the input and returns a random output b_j (that is obtained of the input a_i with the conditional probabilities q_{ij}). In this setting, the sender is free to choose arbitrarily the symbols on the channel's input. The output symbols b_j delivered to the receiver are typically interpreted as a "noisy" version of the sent symbols a_i. By controlling the sequence of symbols $(a_{i_1}a_{i_2}a_{i_3}\ldots)$ given to the input of the channel, the sender suppose to influence the sequence of symbols $(b_{i_1}b_{i_2}b_{i_3}\ldots)$ obtained on the output. The correlation between the input and output symbols permits to transfer via the channel some "useful" information.

This class of channels is called *memoryless* because for each unit of time, the distribution on the possible outputs b_j depends only on the inputs symbol a_i sent at this very moment, with no contribution of symbols transmitted by the channel earlier or later.

Standard examples of random noisy channels :

(1) Binary symmetric channel : the input and the output alphabets both consist of 0 and 1 ; the output bit is different of the corresponding input bit with probability ϵ (i.e., the channel flips the bit with some fixed probability ϵ). More formally, $A = B = \{0, 1\}$, and

$$\begin{aligned} q_{00} &= 1-\epsilon \\ q_{01} &= \epsilon \\ q_{11} &= 1-\epsilon \\ q_{10} &= \epsilon \end{aligned}$$

(2) Binary non symmetric channel : the input and the output alphabets both consist of 0 and 1 ; zeros are always transmitted without errors, and every one is converted in a zero with some fixed probability ϵ. More formally, $A = B = \{0, 1\}$, and

$$\begin{aligned} q_{00} &= 1 \\ q_{01} &= 0 \\ q_{11} &= 1-\epsilon \\ q_{10} &= \epsilon \end{aligned}$$

(3) Binary erasure channel : the input alphabet is $\{0, 1\}$, the output alphabet is $\{0, 1, ?\}$; zero and one are transmitted without error with a probability

$1-\epsilon$, and with the complement probability ϵ they are "lost", i.e., converted to the symbol " ? ". More formally,

$$\begin{array}{rcl} q_{00} & = & 1-\epsilon \\ q_{01} & = & 0 \\ q_{0?} & = & \epsilon \\ q_{11} & = & 1-\epsilon \\ q_{10} & = & 0 \\ q_{1?} & = & \epsilon \end{array}$$

2. Capacity of a noisy channel. By definition, a discrete memoryless channel provides the conditional probabilities (*). However, the channel itself does not fix the distribution on inputs and outputs. If we choose any distribution on the input,

$$p_1 = \text{Prob}[\text{ input} = a_1\ ], \ldots, p_m = \text{Prob}[\text{ input} = a_m\ ],$$

the conditional probabilities (q_{ij}) give us a joint distribution of probabilities on the input and the output of the channel :

$$\text{Prob}[\text{ input} = a_i \text{ and output} = b_j\ ] = p_i \cdot q_{ij}.$$

We will denote this distribution (α, β) (α stands for an input signal and β stands for the output). We focus on the maximal possible value of the mutual information between α and β.

Definition. The *capacity* of a discrete memoryless channel with transition probabilities (q_{ij}) is defined as

$$R := \max_{(p_1 \ldots p_m)} I(\alpha : \beta)$$

(the maximum is taken over all distributions on the channel's inputs).

In the class we computed the capacity for the binary symmetric channel : if the bits are flipped by the channel with a probability ϵ, then the capacity is $R = 1 - h(\epsilon)$, where

$$h(\epsilon) := \epsilon \log \frac{1}{\epsilon} + (1-\epsilon) \log \frac{1}{1-\epsilon}.$$

3. The operational meaning of the channel capacity. The definition of the channel capacity is justified by the following theorem. It shows how much "useful" information we can transmit via a probabilistic noisy channel.

Theorem [Shannon]. Let R be the capacity of a discrete memoryless channel (with an input alphabet A, an output alphabet B, and with transition probabilities (q_{ij})).

(a) For every $L < R$ and for every $\delta > 0$ and for all large enough k there exist functions of encoding and decoding

$$\mathbf{code}_k : \{0,1\}^k \to A^{\lfloor k/L \rfloor} \text{ and } \mathbf{decode}_k : B^{\lfloor k/L \rfloor} \to \{0,1\}^k$$

such that for every $\bar{m} \in \{0,1\}^k$ the probability of get an error at the end of the chain of transformations

$$\bar{m} = (m_1 \dots m_k) \xrightarrow{\mathbf{code}_k} (a_{i_1} \dots a_{i_n}) \xrightarrow{\text{noisy channel}} (b_{i_1} \dots b_{i_n}) \xrightarrow{\mathbf{decode}_k} \bar{m}' = m_1' \dots m_k' \neq \bar{m}$$

is less than δ.

(b) If $L > R$ then for all $\delta > 0$ and for all large enough k, for all functions of encoding and decoding

$$\mathbf{code}_k : \{0,1\}^k \to A^{\lfloor k/L \rfloor} \text{ and } \mathbf{decode}_k : B^{\lfloor k/L \rfloor} \to \{0,1\}^k$$

for *most* messages $\bar{m} \in \{0,1\}^k$ the probability of get an error at the end of the chain of transformations

$$\bar{m} = (m_1 \dots m_k) \xrightarrow{\mathbf{code}_k} (a_{i_1} \dots a_{i_n}) \xrightarrow{\text{noisy channel}} (b_{i_1} \dots b_{i_n}) \xrightarrow{\mathbf{decode}_k} \bar{m}' = m_1' \dots m_k' \neq \bar{m}$$

is greater than $1 - \delta$.

Speaking informally, this theorem claims that we can use the channel of capacity R to transmit $\approx R$ "useful" bits per one symbol sent through the channel. The probability of the decoding error can be made arbitrarily small, if we encode the data in large enough blocks. It is instructive to compare this theorem with Shannon's theorem on block coding for a channel without noise.

Shannon's theorem characterizes the optimal performance of a noisy channel but it does not give a practical recipe how to achieve this performance.

Exercise 11.1. The *random erasure code* has the two letters alphabet $\{0,1\}$ on the input and the three letter alphabet $\{0,1,?\}$ on the output, with the following conditional probabilities :

$$\begin{array}{l}
\text{Prob[output} = 0 \mid \text{input} = 0 \text{]} = 1 - \epsilon, \\
\text{Prob[output} = 1 \mid \text{input} = 1 \text{]} = 1 - \epsilon, \\
\text{Prob[output} = \ ? \mid \text{input} = 0 \text{]} = \epsilon, \\
\text{Prob[output} = \ ? \mid \text{input} = 1 \text{]} = \epsilon.
\end{array}$$

Compute Shannon's capacity of this channel.

Exercise 11.2. The *asymmetric random binary code* has the two letters alphabet $\{0,1\}$ on the input and the three letter alphabet $\{0,1\}$ on the output, with the following conditional probabilities :

$$\begin{array}{l}
\text{Prob[output} = 0 \mid \text{input} = 0 \text{]} = 1, \\
\text{Prob[output} = 1 \mid \text{input} = 1 \text{]} = 1 - \epsilon, \\
\text{Prob[output} = 0 \mid \text{input} = 1 \text{]} = \epsilon.
\end{array}$$

Compute Shannon's capacity of this channel for $\epsilon = 1/2$.

Exercise 11.3 (optional)**.** Compute Shannon's capacity of the channel from Exercise 2 for an arbitrary ϵ.

Exercise 11.4. Let us choose at random 10 binary strings of length 1000,

$$\begin{array}{lcl} \bar{x}_1 & = & (x_{1,1} \quad x_{1,2} \quad \dots \quad x_{1,1000}) \\ \bar{x}_2 & = & (x_{2,1} \quad x_{2,2} \quad \dots \quad x_{2,1000}) \\ & \vdots & \\ \bar{x}_{10} & = & (x_{10,1} \quad x_{10,2} \quad \dots \quad x_{10,1000}) \end{array}$$

(each bit $x_{i,j}$ is chosen be equal to 0 or 1 with probabilities 1/2). Prove that with a positive probability (and even with a probability > 0.5) the obtained set of words $\{\bar{x}_1, \dots \bar{x}_{10}\}$ is an error correcting code that corrects at least 50 errors, i.e., the Hamming distance between every two words $\bar{x}_i$ and $\bar{x}_j$ is greater than $50 + 50 = 100$.

10.12.2019. Lecture 12. Kolmogorov complexity

Definition. Let $L : \{0,1\}^* \to \{0,1\}^*$ be a (possibly partial) computable function. We define $C_L(x) = \min\{|p| \ : \ L(p) = x\}$. (With the convention that minimum of the empty set is infinity.) The function L is often called *decompressor*, and $C_L(x)$ is the complexity of x with respect to this decompressor.

Definition. Let L_1, L_2 be two computable functions. We say that L_1 is *better* than L_2 as a decompressor, if there exists a number Const such that for all strings $x \in \{0,1\}^*$

$$C_{L_1}(x) \le C_{L_2}(x) + \text{Const}.$$

In the class we proved the following statement :

Theorem. There exists a decompressor (a partial computable function) L_{opt} that is better than any other decompressor, i.e., for all computable L and for all binary strings x

$$C_{L_{opt}}(x) \le C_L(x) + O(1).$$

Such a decompressor L_{opt} is called *optimal.*

Remark. There exist infinitely many optimal decompressors. However, they are equivalent to each other in the following sense. If L_{opt} and L'_{opt} are tow optimal decompressors, that there exists a constant Const such that for all binary strings x

$$|C_{L_{opt}}(x) - C_{L'_{opt}}(x)| \le \text{Const}.$$

We fix some optimal decompressor L_{opt} and denote in what follows

$$C(x) := C_{L_{opt}}(x).$$

The value of $C(x)$ is called *Kolmogorov complexity* of x.

In the class we proved several properties of Kolmogorov complexity :

— there exists a constant c_1 such that for all binary strings x

$$C(x) \le |x| + c_1;$$

— there exists a constant c_2 such that for all binary strings x

$$C(xx) \le |x| + c_2;$$

— there exists a constant c_3 such that for all binary strings x

$$C(xx) \le C(x) + c_3;$$

— there exists a constant c_4 such that for all binary strings x

$$C(\underbrace{xx\dots x}_{n}) \le |x| + 2\log n + c_4;$$

— for every computable function f there exists a constant c_5 such that

$$C(f(x)) \leq C(x) + c_5;$$

— for every n there exists a binary string x of length n such that $C(x) \geq n$;

— there exists a constant c_6, such that for every n for at least 99% fo all binary strings of length n

$$n - c_6 \leq C(x) \leq n + c_6.$$

Lemma. There is no algorithm that finds for every given integer $n > 0$ a binary string x_n such that $C(x_n) > n$.

Proposition 1. The value of Kolmogorov complexity $C(x)$ is not a computable function of x.

Corollary. Every optimal decompressor is a partial (not total!) computable function.

Notation. We fix a computable injective function

$$\text{code} : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$$

and denote $C(x, y) := C(\text{code}(x, y))$. Similarly we define $C(x, y, z) := C(\text{code}(\text{code}(x, y), z))$, and so on.

Proposition 2.

(a) $C(x, y) \leq |x| + |y| + 2 \log |x| + O(1)$

(b) $C(x, y) \leq C(x) + C(y) + 2 \log C(x) + O(1)$

(c) for every number c there exist binary strings x and y such that

$$C(x, y) > C(x) + C(y) + c$$

In other words, we cannot omit the logarithmic term in (b) :

$$C(x, y) \not\leq C(x) + C(y) + O(1).$$

Exercise 12.5. Prove that there exists an integer number Const such that for every pair of binary strings x, y

$$C(x, y) \leq |x| + |y| + \log |x| + \log\log |x| + \log\log\log |x| + 2 \log\log\log\log |x| + \text{Const}.$$

Exercise 12.6. Let x be a binary string of length n with *pn zeros* and $(1-p)n$ *ones*. Prove that

$$C(x) \leq \left(p \log \frac{1}{p} + (1-p) \log \frac{1}{1-p}\right) n + O(\log n).$$

Exercise 12.7. Most statements about Kolmogorov complexity are independent of the choice of the optimal decompressor. In this exercise we ask about properties that do depend on the optimal decompressor.

(a) Does there exist an optimal decompressor L_{opt} such that for all x the value of $C_{L_{opt}}(x)$ is *even*?

(b) Does there exist an optimal decompressor L_{opt} such that for all x the value of $C_{L_{opt}}(x)$ is *a power of* 2?

10.12.2019. Lecture 13. Conditional Kolmogorov complexity and communication complexity.

Definition. Let $L : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$ be a (possibly partial) computable function. We define $C_L(x \mid y) = \min\{|p| \ : \ L(p,y) = x\}$. (With the usual convention that minimum of the empty set is infinity.) The function L is often called *decompressor* with a condition, and $C_L(x \mid y)$ is the complexity of x given y with respect to this decompressor.

Definition. Let L_1, L_2 be two computable functions of two arguments. We say that L_1 is *better* than L_2 as a decompressor, if there exists a number Const such that for all strings $x, y \in \{0,1\}^*$

$$C_{L_1}(x \mid y) \leq C_{L_2}(x \mid y) + \text{Const}.$$

Theorem. There exists a decompressor (a partial computable function) L_{opt} that is better than any other decompressor, i.e., for all computable L and for all binary strings x

$$C_{L^c_{opt}}(x \mid y) \leq C_L(x \mid y) + O(1).$$

Such a decompressor L^c_{opt} is called *optimal.*

We fix some optimal decompressor L^c_{opt} and denote in what follows

$$C(x \mid y) := C_{L_{opt}}(x \mid y).$$

The value of $C(x)$ is called *conditional Kolmogorov complexity* of x given y.

In the class we proved several properties of Kolmogorov complexity :

— $C(x \mid y) \leq C(x) + O(1)$;

— $C(x \mid \epsilon) = C(x) + O(1)$;

— for every computable function f there exists an integer number const_f such that
$$C(f(x) \mid x) \leq \text{const}_f;$$

— there exists a number const > 0 such that for every n for at least 99% fo all binary strings x of length n and for all y (of any length)
$$n - \text{const} \leq C(x \mid y) \leq n + \text{const}.$$

Definition. Information in x on y is defined as $I(x : y) := C(y) - C(y \mid x)$.

Theorem [Kolmogorov–Levin]. There exist integer numbers c_1, c_2 such that for all binary strings x, y

$$\big|C(x,y) - C(x) - C(y \mid x)\big| \leq c_1 \log N + c_2,$$

where $N = C(x) + C(y)$.

Corollary. The mutual information is symmetric up to an additive logarithmic term :

$$I(x:y) = I(x:y) + O(\log N),$$

where $N = C(x) + C(y)$.

In the class we proved the following remark : the additive logarithmic term in the theorem of Kolmogorov–Levin cannot be omitted. This is not an artifact of the proof ; the mutual information is indeed symmetric only with a logarithmic "correction term."

Applications : In the class we discussed two arguments based on Kolmogorov complexity. In the first example we used Kolmogorov complexity to prove the classical theorem : there are infinitely many prime numbers. In the second example we proved that a Turing machine with one tape cannot duplicate the given input word faster than in time $\Omega(n)$ (where n is the length of the input).

Communication complexity. In the class we introduced the notion of *deterministic communication protocol* for two parties (Alice and Bob) and defined *communication complexity* of a function

$$f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^m.$$

We proved that for the predicate $\mathrm{EQ}_n : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^m$ defined as

$$\mathrm{EQ}_n(x,y) = \begin{cases} 1, & \text{if } x = y \\ 0, & \text{if } x \neq y \end{cases}$$

the deterministic communication complexity is equal to $n+1$.

Then we introduced the model of *randomized* communication and constructed for the predicate EQ_n a randomized communication protocol with communication complexity $O(\log n)$, which gives for each pair of inputs x, y the right answer with a probability > 0.99.