Elements of Information Theory
Cover and Thomas

Eu-Jin Goh
eujin@cs.stanford.edu

May 29, 2003

1 Introduction and Preview

2 Entropy, Relative Entropy, and Mutual Information

2.1 Entropy

Entropy is a measure of the average uncertainty of a random variable. It is also a measure of the amount of information required on the average to describe the random variable.

Definition (Entropy). Let X be a discrete random variable with alphabet $\mathcal{X}$ and probability mass function $p(x) = \Pr[X = x]$ where $x \in \mathcal{X}$. The *entropy* $H(X)$ of X is defined by

$$H(X) = -\sum_{x \in \mathcal{X}} p(x) \log p(x).$$

If the log is base 2, then entropy is expressed in bits. For example, the entropy of a fair coin toss is 1 bit. If the log is base e, then entropy is expressed in nats.

Recall that the expectation of a random variable $g(X)$ is defined as

$$Eg(X) = \sum_{x \in \mathcal{X}} g(x)p(x).$$

We can interprete the entropy of X as the expected value of $\log(1/p(X))$, i.e. $H(X) = E\log(1/p(X))$.

Lemma 2.1.1. $H(X) \geq 0$.

Lemma 2.1.2. $H_b(X) = (\log_b a)H_a(X)$.

2.2 Joint and Conditional Entropy

Definition (Joint Entropy). The *joint entropy* of a pair of discrete random variables (X, Y) with a joint distribution $p(x, y)$ is defined by

$$H(X, Y) = -\sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{B}} p(x, y) \log p(x, y).$$

Definition (Conditional Entropy). If $(X, Y) \sim p(x, y)$, then the *conditional entropy* $H(Y|X)$ is defined as

$$\begin{aligned}
H(Y|X) &= \sum_{x \in \mathcal{X}} p(x) H(Y|X = x) \\
&= -\sum_{x \in \mathcal{X}} p(x) \sum_{y \in \mathcal{B}} p(y|x) \log p(y|x) \\
&= -\sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{B}} p(x, y) \log p(y|x) \\
&= E_{p(x,y)} \log p(Y|X).
\end{aligned}$$

Theorem 2.2.1 (Joint Entropy Chain Rule). $H(X, Y) = H(X) + H(Y|X)$.

Corollary. $H(X, Y|Z) = H(X|Z) + H(Y|X, Z)$.

2.3 Relative Entropy and Mutual Information

Relative entropy $D(p\|q)$ is a measure of the distance between two distributions p and q. It is also a measure of the inefficiency of assuming that the distribution is q when it is actually p.

For example, if we knew the true distribution of a random variable to be p, then we can construct a code with average description length $H(p)$. If we instead used the code for a distribution q, we would need $H(p) + D(p\|q)$ bits on the average to describe the random variable.

Definition (Relative Entropy). The *relative entropy or Kullback-Leibler distance* between two probability mass functions $p(x)$ and $q(x)$ is defined as

$$D(p\|q) = \sum_{x \in \mathcal{X}} p(x) \log \frac{p(x)}{q(x)} = E_p \log \frac{p(X)}{q(X)}.$$

Note that in general $D(p\|q) \neq D(q\|p)$.

Mutual information is a measure of the amount of information one random variable contains about another. It is the reduction in the uncertainty of one random variable due to the knowledge of another.

Definition (Mutual Information). Consider two random variables X and Y with a joint probability mass function $p(x,y)$ and marginal probability mass functions $p(x)$ and $p(y)$. The *mutual information* $I(X;Y)$ is the relative entropy between the joint distribution and product distribution $p(x)p(y)$, i.e.

$$\begin{aligned} I(X;Y) &= \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{B}} p(x,y)\log\frac{p(x,y)}{p(x)p(y)} \\ &= D(p(x,y)\|p(x)p(y)) \\ &= E_{p(x,y)}\log\frac{p(X,Y)}{p(X)p(Y)} \end{aligned}$$

2.4 Relation between Entropy and Mutual Information

Theorem 2.4.1 (Mutual Information and Entropy).

$$\begin{aligned} I(X;Y) &= H(X)-H(X|Y) \\ I(X;Y) &= H(Y)-H(Y|X) \\ I(X;Y) &= H(X)+H(Y)-H(X,Y) \\ I(X;Y) &= I(Y;X) \\ I(X;X) &= H(X) \end{aligned}$$

2.5 Chain Rules for Entropy, Relative Entropy, and Mutual Information

The entropy of a collection of random variables is the sum of the conditional entropies.

Theorem 2.5.1 (Entropy Chain Rule). *Let $X_1,\ldots,X_n$ be drawn according to $p(x_1,\ldots,x_n)$. Then,*

$$H(X_1,\ldots,X_n)=\sum_{i=1}^{n} H(X_i|X_{i-1},\ldots,X_1).$$

The conditional mutual information is the reduction in the uncertainty of X due to knowledge of Y when Z is given.

Definition (Conditional Mutual Information). The *conditional mutual information* of random variables X and Y given Z is defined by

$$\begin{aligned} I(X;Y|Z) &= E_{p(x,y,z)}\frac{p(X,Y|Z)}{p(X|Z)p(Y|Z)} \\ &= H(X|Z)-H(X|Y,Z). \end{aligned}$$

Theorem 2.5.2 (Information Chain Rule).

$$I(X_1,\ldots,X_n;Y)=\sum_{i=1}^{n} I(X_i;Y|X_{i-1},\ldots,X_1).$$

Definition (Conditional Relative Entropy). The conditional relative entropy $D(p(x|y)\|q(x|y))$ is the average of the relative entropies between the conditional probability mass functions $p(x|y)$ and $q(x|y)$ averaged over the probability mass function $p(x)$. i.e.

$$\begin{aligned} D(p(x|y)\|q(x|y)) &= \sum_{x\in\mathcal{X}} p(x)\sum_{y\in\mathcal{B}} p(x|y)\log\frac{p(x|y)}{q(x|y)} \\ &= E_{p(x,y)}\log\frac{p(X|Y)}{q(X|Y)} \end{aligned}$$

Theorem 2.5.3 (Relative Entropy Chain Rule).

$$D(p(x,y)\|q(x,y)) = D(p(y)\|q(y))+D(p(x|y)\|q(x|y)).$$

2.6 Jensen's Inequality and its Consequences

Definition (Convex Functions). A function $f(x)$ is *convex* over an interval (a,b) if for every $x_1,x_2\in(a,b)$ and $0\le\lambda\le 1$,

$$f(\lambda x_1+(1-\lambda)x_2)\le\lambda f(x_1)+(1-\lambda)f(x_2).$$

A function is said to be *strictly convex* if equality holds only if $\lambda=0$ or 1.

A function is convex if it always lies below any chord.

Definition (Concave Functions). A function f is *concave* if $-f$ is convex.

A function is concave if it always lies above any chord.

Theorem 2.6.1. *If the function f has a non-negative (positive) second derivative everywhere, then the function is convex (strictly convex).*

Theorem 2.6.2 (Jensen's Inequality). *If f is a convex function, and X a random variable, then $Ef(X)\ge f(EX)$. If f is strictly convex, then equality implies that that $X=EX$ with probability 1, i.e. X is constant.*

Note that for discrete distributions,

$$\sum_{i=1}^{k} p_i f(x_i)=Ef(X)\ge f(EX)=f(\sum_{i=1}^{k} p_i x_i).$$

Theorem 2.6.3 (Information Inequality). *Let $p(x), q(x), x \in \mathcal{X}$ be two probability mass functions. Then,*

$$D(p\|q) \geq 0$$

with equality if and only if $p(x) = q(x)$ for all x.

Corollary (Non-negativity of Mutual Information). For any two random variables X and Y,

$$I(X;Y) \geq 0$$

with equality if and only if X and Y are independent.

Corollary. For any two random variables X and Y,

$$D(p(y|x)\|q(y|x)) \geq 0$$

with equality if and only if $p(y|x) = q(y|x)$ for all y and x with $p(x) > 0$.

Corollary. For any two random variables X and Y,

$$I(X;Y|Z) \geq 0$$

with equality if and only if X and Y are conditionally independent given Z.

The uniform distribution over the range $\mathcal{X}$ is the maximum entropy distribution over this range. Any random variable within this range has an entropy no greater than $\log |\mathcal{X}|$.

Theorem 2.6.4. *Let $|\mathcal{X}|$ be the number of elements in the range of a random variable X. Then,*

$$H(X) \leq \log |\mathcal{X}|$$

with equality if and only if X has a uniform distribution over $\mathcal{X}$.

Theorem 2.6.5 (Conditioning Reduces Entropy). *For any two random variables X and Y,*

$$H(X|Y) \leq H(X)$$

with equality if and only if X and Y are independent.

Intuitively, knowing another random variable Y can only reduce the uncertainty in a random variable X. Note that the theorem is only true on the average.

Theorem 2.6.6 (Independence Bound on Entropy). *Let $X_1, \ldots, X_n$ be drawn according to $p(x_1, \ldots, x_n)$. Then,*

$$H(X_1, \ldots, X_n) \leq \sum_{i=1}^{n} H(X_i)$$

with equality if and only if the X_i are independent.

2.7 The Log Sum Inequality and its Applications

2.8 Data Processing Inequality

The data processing inequality shows that no manipulation of data can improve the inferences that can be made from the data.

Definition (Ordered Markov Chain). The random variables X, Y, Z form a *Markov Chain in that order* (denoted by $X \to Y \to Z$) if the conditional distribution of Z depends only on Y and is conditionally independent of X. Specifically, X, Y and Z form a Markov chain $X \to Y \to Z$ if the joint probability mass function can be written as

$$p(x, y, z) = p(x)p(y|x)p(z|y).$$

Lemma. The random variables X, Y, Z form a ordered Markov chain if and only if X and Z are conditionally independent given Y, i.e. $p(x, z|y) = p(x|y)p(z|y)$.

Lemma. $X \to Y \to Z$ implies $Z \to Y \to X$.

Lemma. If $Z = f(y)$, then $X \to Y \to Z$.

No processing on Y (deterministic or random) can increase the information that Y contains about X.

Theorem 2.8.1 (Data Processing Inequality). *If $X \to Y \to Z$, then $I(X;Z) \leq I(X;Y)$.*

Functions of Y cannot increase the information about X.

Corollary. If $Z = g(Y)$, then $I(X, g(Y)) \leq I(X;Y)$.

The dependence of X and Y is decreased or remains unchanged by the observation of a "downstream" variable Z.

Corollary. If $X \to Y \to Z$, then $I(X;Y|Z) \leq I(X;Y)$.

2.9 The Second Law of Thermodynamics

2.10 Sufficient Statistics

2.11 Fano's Inequality

Fano's inequality relates the probability of error in guessing a random variable X to its conditional entropy $H(X|Y)$ where Y is correlated to X.

We know that $H(X|Y) = 0$ if and only if X is a function of Y. So we can estimate X from Y with 0 error probability if and only if $H(X|Y) = 0$. Fano's inequality quantifies our intuition that we can expect to estimate X with low error probability only if $H(X|Y)$ is small.

Suppose we want to estimate a random variable X with a distribution $p(x)$. We observe a random variable Y that is related to X by the conditional distribution $p(y|x)$. We estimate X from Y using a function g, where $g(Y) = \hat{X}$. We want to bound the probability that $\hat{X} \neq X$. Define error probability as

$$P_e = \Pr\{\hat{X} \neq X\}.$$

Also observe that $X \to Y \to \hat{X}$.

Theorem 2.11.1 (Fano's Inequality).

$$H(P_e) + P_e \log(|\mathcal{X}| - 1) \geq H(X|Y)$$

which can be weakened to

$$1 + P_e \log(|\mathcal{X}|) \geq H(X|Y).$$

3 The Asymptotic Equipartition Property

The AEP is the analog of the law of large numbers. The AEP states that for independent, identically distributed (i.i.d.) random variables $X_1, \ldots, X_n$, $(1/n)\log(1/p(X_1, \ldots, X_n))$ is close to the entropy H. Thus, the probability assigned to an observed sequence is close to 2^{-nH}.

We can divide the set of all sequences into two sets, the typical set where the sample entropy is close to the true entropy, and the non-typical set which contains the other sequences.

3.1 The AEP

Theorem 3.1.1 (AEP). *If $X_1, \ldots, X_n$ are i.i.d.$\sim p(x)$, then*

$$-(1/n)\log p(X_1, \ldots, X_n) \to H(X)$$

in probability.

Definition (Typical Set). The *typical set* $A_\epsilon^{(n)}$ with respect to $p(x)$ is the set of sequences $(x_1, \ldots, x_n) \in \mathcal{X}^n$ such that

$$2^{-n(H(X)+\epsilon)} \leq p(x_1, \ldots, x_n) \leq 2^{-n(H(X)-\epsilon)}.$$

The next theorem shows that the typical set has probability nearly 1, all elements of the typical set are nearly equiprobable, and the number of elements in the typical set is nearly 2^{nH}.

Theorem 3.1.2 (AEP Consequences).

1. *If $(x_1, \ldots, x_n) \in A_\epsilon^{(n)}$, then*

   $$H(X) - \epsilon \leq -(1/n)\log p(x_1, \ldots, x_n) \leq H(X) + \epsilon.$$

2. $\Pr\{A_\epsilon^{(n)}\} > 1 - \epsilon$ *for sufficiently large n.*

3. $|A_\epsilon^{(n)}| \leq 2^{n(H(X)+\epsilon)}$ *where $|A|$ denotes the number of elements in the set A.*

4. $|A_\epsilon^{(n)}| \geq (1-\epsilon)2^{n(H(X)-\epsilon)}$ *for sufficiently large n.*

3.2 Data Compression

In any set of sequences $\mathcal{X}^n$, we can divide the set into two, the typical set $A_\epsilon^{(n)}$ and its complement.

Each sequence in $A_\epsilon^{(n)}$ can be represented by giving the index of the sequence in the set. Since there are less than $2^{n(H+\epsilon)}$ sequences in $A_\epsilon^{(n)}$, the indexing requires $n(H+\epsilon)+1$ bits. We prefix these sequences by a 0. So each sequence in $A_\epsilon^{(n)}$ requires less than $n(H+\epsilon)+2$ bits to represent.

Each sequence not in $A_\epsilon^{(n)}$ can be represented (brute force enumeration) in less than $n \log |\mathcal{X}| + 1$ bits. Prefixing these indices by 1, we have a code for all of $\mathcal{X}^n$.

The following theorem tells us the expected length of this code.

Theorem 3.2.1. *Let X^n be i.i.d.$\sim p(x)$. Also let $\epsilon > 0$. Then there exists a code which maps sequences x^n of length n into binary strings such that the mapping is 1-1 (and therefore invertible) and*

$$E[l(X^n)] \leq nH(X) + n\epsilon$$

for sufficiently large n.

Hence, we can represent sequences X^n using $nH(X)$ bits on the average.

3.3 High Probability Sets and the Typical Set

This section shows that the typical set has essentially the same number of elements as the smallest set which contains most of the probability.

4 Entropy Rates of a Stochastic Process

The AEP shows that $nH(X)$ bits are enough to describe n i.i.d. random variables. We examine the case where the random variables are dependent and the case where they form a stationary process.

In this section, we show that the entropy $H(X_1, \ldots, X_n)$ for stationary processes grows linearly with n.

4.1 Markov Chains

A stochastic process is an indexed sequence of random variables and is characterised by the joint probability mass functions $\Pr\{(X_1, \ldots, X_n) = (x_1, \ldots, x_n)\} = p(x_1, \ldots, x_n)$ where $(x_1, \ldots, x_n) \in \mathcal{X}^n$ for $n \in \mathbb{Z}^+$.

Definition (Stationary Stochastic Process). A stochastic process is said to be *stationary* if the joint probability distribution of any subset of the sequence of random variables is invariant with respect to shifts in the time index, i.e.

$$\begin{aligned}&\Pr\{X_1 = x_1, \ldots, X_n = x_n\} = \\ &\qquad \Pr\{X_{1+l} = x_{1+l}, \ldots, X_{n+l} = x_{n+l}\}\end{aligned}$$

for every shift l and $x_1, \ldots, x_n \in \mathcal{X}$.

Intuitively, a Markov process is a stochastic process where each random variable depends on its predecessor and is conditionally independent of all other preceding random variables.

Definition (Markov Chain). A discrete stochastic process is a *Markov Chain* if

$$\begin{aligned}&\Pr\{X_{n+1} = x_{n+1} \mid X_n = x_n, \ldots, X_1 = x_1\} \\ &\quad = \Pr\{X_{n+1} = x_{n+1} \mid X_n = x_n\}\end{aligned}$$

for $n \in \mathbb{Z}^+$ and $x_1, \ldots, x_n \in \mathcal{X}$.

For a Markov Chain, the joint pmf can be written

$$p(x_1, x_2, \ldots, x_n) = p(x_1)p(x_2|x_1) \ldots p(x_n|x_{n-1}).$$

Definition (Time Invariant Markov Chain). A Markov Chain is *time invariant* if the conditional probability $p(x_{n+1}|x_n)$ does not depend on n, i.e.

$$\Pr\{X_{n+1} = b \mid x_{n+1} = a\} = \Pr\{X_2 = b \mid X_1 = a\}.$$

for $a, b \in \mathcal{X}$.

We will assume that a Markov Chain is time invariant unless otherwise stated.

Definition (State of a Markov Chain). If $\{X_i\}$ is a Markov Chain, then X_n is its *state* at time n.

A time invariant Markov Chain is characterized by its initial state and its probability transition matrix $P = [P_{ij}]$ where $P_{ij} = \Pr\{X_{n+1} = j \mid X_n = i\}$.

Definition (Irreducible Markov Chain). An *irreducible Markov Chain* is a chain where it is possible to go from any state of the chain to any other state with positive probability in a finite number of steps.

If the pmf of the random variable at time n is $p(x_n)$, then the pmf at time $n+1$ is

$$p(x_{n+1}) = \sum_{x_n} p(x_n) P_{x_n x_{n+1}}.$$

Definition (Stationary Distribution). A *stationary distribution* is a distribution on states such that the distribution at time $n+1$ is the same as the distribution at time n.

If the initial state of a Markov Chain is drawn according to a stationary distribution, then the Markov Chain forms a stationary process.

4.2 Entropy Rate

Intuitively, the entropy rate is the rate of growth of entropy of a sequence of n random variables as n grows.

Definition (Entropy Rate). The *entropy rate* of a stochastic process $\{X_i\}$ is defined by

$$H(\mathcal{X}) = \lim_{n\to\infty} \frac{1}{n} H(X_1, \ldots, X_n)$$

when the limit exists.

$H(\mathcal{X})$ is the per symbol entropy of the n random variables. Consider the related quantity for entropy rate,

$$H'(\mathcal{X}) = \lim_{n\to\infty} H(X_n|X_{n-1}, \ldots, X_1)$$

when the limit exists. $H'(\mathcal{X})$ is the conditional entropy of the last random variable given the past.

Theorem (Entropy Rate for i.i.d. Random Variables). If $X_1, \ldots, X_n$ are i.i.d. random variables, then $H(\mathcal{X}) = H(X_1)$.

Theorem 4.2.1. *For a stationary stochastic process, the limits for $H(\mathcal{X})$ and $H'(\mathcal{X})$ exist and are equal, i.e. $H(\mathcal{X}) = H'(\mathcal{X})$.*

The following two theorems are used to prove Theorem 4.2.1.

Theorem 4.2.2. *For a stationary stochastic process, $H(X_n|X_{n-1}, \ldots, X_1)$ is decreasing in n and has a limit $H'(\mathcal{X})$.*

Theorem 4.2.3 (Cesaro mean). *If $a_n \to a$ and $b_n = (1/n)\sum_{i=1}^{n} a_i$, then $b_n \to a$.*

For a stationary Markov Chain, $H(\mathcal{X}) = H'(\mathcal{X}) = \lim H(X_n|X_{n-1}, \ldots, X_1) = \lim H(X_n|X_{n-1}) = H(X_2|X_1)$. The following theorem gives the entropy rate for a stationary Markov Chain.

Theorem 4.2.4. *Let $\{X_i\}$ be a stationary Markov Chain with stationary distribution μ and transition matrix P. Then the entropy rate is*

$$H(\mathcal{X}) = H(X_2|X_1) = -\sum_{ij} \mu_i P_{ij} \log P_{ij}.$$

4.3 Entropy Rate of a Random Walk on a Weighted Undirected Graph

Let W_i be the total weight of edges emanating from node i. Let W be the sum of weights of all the edges. Then, the stationary distribution of the graph is

$$\nu_i = W_i/2W.$$

Let E_i be the number of edges emanating from node i and E is the total number of edges in the graph. If all the edges have equal weight, then the stationary distribution on node i is $E_i/2E$ and the entropy rate of a random walk on the graph is

$$H(\mathcal{X}) = \log(2E) - H(E_1/2E, \ldots, E_n/2E).$$

5 Data Compression

5.1 Examples of Codes

Definition (Source Code). A *source code* C for a random variable X is a mapping from $\mathcal{X}$ to $\mathcal{D}^*$, the set of finite length strings of symbols from a D-ary alphabet. Let $C(x)$ denote the codeword corresponding to x and let $l(x)$ denote the length of $C(x)$.

Definition (Expected Source Code Length). The *expected length* $L(C)$ of a source code $C(x)$ for a random variable X with probability mass function $p(x)$ is given by

$$L(C) = \sum_{x \in \mathcal{X}} p(x) l(x)$$

where $l(x)$ is the codeword associated with x.

Definition (Non-singular Codes). A code is *non-singular* if every element of $\mathcal{X}$ maps into a different string in $\mathcal{D}^*$, i.e. if $x_i \neq x_j$, then $C(x_i) \neq C(x_j)$.

Non-singularity implies an unambiguous description of a single value of X. But sequences of symbols might still be ambiguous.

Definition (Code Extension). The *extension* C^* of a code C is the mapping from finite length strings of $\mathcal{X}$ to finite length strings of $\mathcal{D}$ defined by

$$C(x_1 \ldots x_n) = C(x_1) \ldots C(x_n)$$

where $C(x_1) \ldots C(x_n)$ indicated concatenation of the corresponding codewords.

Definition (Uniquely Decodable Codes). A code is *uniquely decodable* if its extension is non-singular.

Definition (Instantaneous Codes). An *instantaneous code* is a code where no codeword is a prefix of any other codeword.

Remark. Instantaneous codes $\subset$ Uniquely Decodable codes $\subset$ Non-singular codes.

5.2 Kraft Inequality

The set of codeword lengths possible for instantaneous codes is limited by the Kraft inequality.

Theorem 5.2.1 (Kraft Inequality). *For any instantaneous code over an alphabet of size D, the codeword lengths $l_1, \ldots, l_m$ must satisfy the inequality*

$$\sum_i D^{-l_i} \leq 1.$$

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

Theorem 5.2.2 (Extended Kraft Inequality). *For any countably infinite set of codewords that form a prefix code, the codeword lengths satisfy the inequality*

$$\sum_{i=1}^{\infty} D^{-l_i} \leq 1.$$

Conversely, given any $l_1, \ldots$ satisfying the extended Kraft inequality, we can construct an instantaneous code with these lengths.

5.3 Optimal Codes

To minimize the codeword length for a codeword set, we must choose lengths for each codeword such that $l_i = -log_D p_i$. This coding scheme is called a Shannon code.

Theorem 5.3.1. *The expected length L of any instantaneous D-ary code for a random variable X is greater than or equal to the entropy $H_D(X)$, i.e.*

$$L \geq H_D(X) = -\sum p_i \log_D p_i$$

with equality if and only if $D^{-l_i} = p_i$.

Definition (D-adic). A probability distribution is *D-adic* with respect to D if each of the probabilities is equal to D^{-n} for some n.

5.4 Bounds on the Optimal Codelength

The optimal code has codeword lengths such that $l_i = -log_D p_i$. However, $-log_D p_i$ might not be an integer so we can round it up to give integer word lengths. The new set of codeword lengths still satisfy the Kraft inequality since

$$\sum D^{-\lceil -\log p_i \rceil} \leq \sum D^{\log p_i} = \sum p_i = 1.$$

The new set of codeword lengths satisfies

$$\log_D(1/p_i) \leq l_i < \log_D(1/p_i) + 1$$

which summing over i and multiplying by p_i gives

$$H_D(X) \leq L < H_D(X) + 1.$$

Since the optimal code can only be better than this code, we have the following theorem.

Theorem 5.4.1. *Let $l_1^*, \ldots, l_m^*$ be the optimal codeword lengths for a source distribution* $\mathbf{p}$ *and a D-ary alphabet. Also let $L^* = \sum p_i l_i^*$ be the associated expected length of the optimal code. Then*

$$H_D(X) \leq L^* < H_D(X) + 1.$$

The theorem gives an overhead of at most 1 bit per symbol because $-log_D p_i$ might not be an integer. We can amortize the overhead by spreading it out over n symbols which can be considered as a single symbol from $\mathcal{X}^n$. The expected codeword length per symbol is

$$\begin{aligned} L_n &= (1/n)\sum p(x_1, \ldots, x_n) l(x_1, \ldots, x_n) \\ &= (1/n) El(X_1, \ldots, X_n). \end{aligned}$$

Applying the entropy bounds to this code, we have

$$H(X_1, \ldots, X_n) \leq nL_n < H(X_1, \ldots, X_n) + 1$$

which gives

$$H(X) \leq L_n < H(X) + (1/n)$$

if $X_1, \ldots, X_n$ are i.i.d.. Hence, using large blocks allows an expected code length per symbol arbitrary close to the entropy.

If the stochastic process $X_1, \ldots, X_n$ is not i.i.d., we can still derive some bounds if it is stationary.

Theorem 5.4.2. *The minimum expected codeword length per symbol satisfies*

$$H(X_1, \ldots, X_n)/n \leq L_n^* < H(X_1, \ldots, X_n)/n + 1/n.$$

If $X_1, \ldots, X_n$ is stationary, $L_n^* \to H(\mathcal{X})$ where $H(\mathcal{X})$ is entropy rate of the process.

The following theorem shows that using the wrong distribution (p) for a code that is designed for another distribution (q) incurs a penalty of $D(p\|q)$.

Theorem 5.4.3. *The expected length under $p(x)$ of the code assignment $l(x) = \lceil \log(1/q(x)) \rceil$ satisfies*

$$H(p) + D(p\|q) \leq E_p l(X) < H(p) + D(p\|q) + 1.$$

5.5 Kraft Inequality for Uniquely Decodable Codes

A surprising result is that when compared to the class of instantaneous codes, the class of uniquely decodable codes does not increase the set of possible codeword lengths.

Theorem 5.5.1 (McMillan Inequality). *The codeword lengths of any uniquely decodable code must satisfy the Kraft inequality $\sum D^{-l_i} \leq 1$. Conversely, given a set of codeword lengths that satisfy this inequality, it is possible to construct a uniquely decodable code with these codeword lengths.*

Corollary. A uniquely decodable code for an infinite source alphabet $\mathcal{X}$ also satisfies the Kraft inequality.

The theorem and corollary shows that the set of codeword lengths is the same for uniquely decodable and instantaneous codes. Hence, the optimal codeword length bounds derived for instantaneous codes also holds for uniquely decodable codes.

5.6 Huffman Codes

Merge the two lowest probability symbols together and assign both of them a prefix symbols of 0 and 1. Continue in this manner assigning additional prefix symbols till there is only one merged symbol.

5.7 Comments on Huffman Codes

5.8 Optimality of Huffman Codes

5.9 Shannon-Fano-Elias Coding

5.10 Arithmetic Coding

5.11 Competitive Optimality of the Shannon Code

5.12 Generation of Discrete Distributions from Fair Coins

We consider the question of how many fair coin flips does it take to generate a random variable drawn according to some specified probability mass function.

Lemma 5.12.1. *Consider a probability distribution on the leaves of any complete tree such that the probability of a leaf at depth k is 2^{-k}. Then the expected depth of the tree is equal to the entropy of the distribution.*

Theorem 5.12.2. *For any algorithm generating a random variable X, the expected number of fair bits used is greater than the entropy $H(X)$.*

Corollary 5.12.3. *Let the random variable X have a d-adic distribution. The optimal algorithm to generate X from fair coin flips requires an expected number of coin tosses equal to the entropy.*

If we wish to generate distributions that are not d-adic, then we should split any probability that is not of the form 2^{-k} into atoms of this form. These atoms be alloted to leaves on the tree. To minimize the expected depth of the tree, we should use atoms with the largest possible probability.

For example, let the random variable X take on the value a with probability $2/3$ and b with probability $1/3$. The binary expansion of these probabilities are

$$\begin{aligned} 2/3 &= 0.10101010\ldots \\ 1/3 &= 0.01010101\ldots \end{aligned}$$

Hence, the atoms for the expansion are

$$\begin{aligned} 2/3 &\rightarrow (1/2, 1/8, 1/32, \ldots) \\ 1/3 &\rightarrow (1/4, 1/16, 1/64, \ldots). \end{aligned}$$

Theorem 5.12.4. *The expected number of fair bits ET required by the optimal algorithm to generate a random variable is given by*

$$H(X) \leq ET < H(X) + 2.$$

6 Gambling and Data Compression

6.1 The Horse Race

Let b_i be the fraction of a gambler's wealth invested in horse i where $b_i \geq 0$ and $\sum b_i = 1$. If horse i wins the race, the gambler receives o_i times the amount of wealth bet on horse i.

Definition (Wealth Relative). The *wealth relative* $S(X) = b(X)o(X)$ is the factor by which the gambler's wealth grows if horse X wins the race.

Let S_n be the gambler's wealth after n races. Then

$$S_n = \prod_{i=1}^{n} S(X_i).$$

Definition (Doubling Rate). The *doubling rate* of a horse race is

$$W(\mathbf{b}, \mathbf{p}) = E(\log S(X)) = \sum_{k=1}^{m} p_k \log b_k o_k.$$

Theorem 6.1.1. *Let the race outcomes $X_1, \ldots, X_n$ be i.i.d. drawn from a distribution $p(x)$. Then the wealth of the gambler using betting strategy $\mathbf{b}$ grows exponentially at rate $W(\mathbf{b}, \mathbf{p})$, i.e.*

$$S_n = 2^{nW(\mathbf{b}, \mathbf{p})}.$$

Definition (Optimum Doubling Rate). The optimum doubling rate $W^*(\mathbf{p})$ is the maximum doubling rate over all choices of the portfolio $\mathbf{b}$, i.e.

$$W^*(\mathbf{p}) = \max_{b} W(\mathbf{b}, \mathbf{p}).$$

Theorem 6.1.2 (Proportional Gambling is log optimal). *The optimum doubling rate is given by*

$$W^*(\mathbf{p}) = \sum p_i \log o_i - H(\mathbf{p})$$

and is achieved by the proportional gambling scheme $\mathbf{b}^* = \mathbf{p}$.

Consider the case where odds are fair with respect to some distribution, i.e. $\sum o_i = 1$. Let $r_i = 1/o_i$. Then

$$\begin{aligned} W(\mathbf{b}, \mathbf{p}) &= \sum p_i \log b_i o_i \\ &= \sum p_i \log((b_i/p_i)(p_i/r_i)) \\ &= D(\mathbf{p}\|\mathbf{r}) - D(\mathbf{p}\|\mathbf{b}). \end{aligned}$$

We can interpret the doubling rate as the difference between the distance of the bookie's estimate from the true distribution and the distance of the gambler's estimate from the true distribution.

When the odds are m for 1 on each horse, the optimum doubling rate is $W^*(\mathbf{p}) = D(\mathbf{p}||(1/m)) = \log m - H(\mathbf{p})$, which gives the following theorem.

Theorem 6.1.3 (Conservation Theorem). *For uniform fair odds,*

$$W^*(\mathbf{p}) + H(\mathbf{p}) = \log m.$$

6.2 Gambling and Side Information

Let Y be the side information a gambler has about the horse race. Let $b(x|y) \geq 0$ and $\sum_x b(x|y) = 1$ be an arbitrary conditional betting strategy.

Let the conditional doubling rate be

$$W(X|Y) = \sum_{x,y} p(x,y) \log b(x|y) o(x).$$

The proof that proportional betting is optimal for the unconditional doubling rate can also be used to show that conditionally proportional gambling is the optimal betting strategy to maximize $W(X|Y)$. Hence,

$$W^*(X|Y) = \sum p(x) \log o(x) - H(X|Y).$$

Theorem 6.2.1. *Let $\Delta W = W^*(X|Y) - W^*(X)$. The increase ΔW in doubling rate due to side information Y for a horse race X is $\Delta W = I(X;Y)$.*

6.3 Dependent Horse Races and Entropy Rate

6.4 The Entropy of English

6.5 Data Compression and Gambling

6.6 Gambling Estimate of the Entropy of English

7 Kolmogorov Complexity

7.1 Models of Computation

Universal Turing machines.

7.2 Definitions and Examples

Definition (Kolmogorov Complexity). The *Kolmogorov Complexity* $K_{\mathcal{U}}(x)$ of a string x with respect to the universal computer $\mathcal{U}$ is defined as

$$K_{\mathcal{U}}(x) = \min_{p:\mathcal{U}(p)=x} l(p)$$

which is the minimum length over all programs that praint x and halts. Thus $K_{\mathcal{U}}(x)$ is the shortest description length of x over all descriptions interpreted by $\mathcal{U}$.

This definition does not take into account the length of x. If we assume the computer already knows the length of x, then we can define the *conditional Kolmogorov complexity* knowing $l(x)$ as

$$K_{\mathcal{U}}(x|l(x)) = \min_{p:\mathcal{U}(p,l(x))=x} l(p).$$

Theorem 7.2.1 (Universality of Kolmogorov Complexity). *If $\mathcal{U}$ is a universal computer, then for any other computer $\mathcal{A}$,*

$$K_{\mathcal{U}}(x) \leq K_{\mathcal{A}}(x) + c_{\mathcal{A}}$$

for all strings $x \in \{0,1\}^*$ where the constant $c_{\mathcal{A}}$ does not depend on x.

This theorem allows to assume a fixed universal computer in the later theorems and definitions.

The next theorem tells us that conditional complexity is less than the length of the sequence. The program simply includes the string x. Note that the program is self delimiting because $l(x)$ is provided.

Theorem 7.2.2 (Upper Bound on Conditional Kolmogorov Complexity).

$$K(x|l(x)) \leq l(x) + c.$$

Without knowledge of the string length, we need a stop symbol or a self punctuating scheme.

Theorem 7.2.3 (Upper Bound on Kolmogorov Complexity).

$$K(x) \leq K(x|l(x)) + 2\log l(x) + c.$$

The next theorem proves that there are very few sequences with low complexity.

Theorem 7.2.4. *The number of strings x with complexity $K(x) < k$ satisfies*

$$|\{x \in \{0,1\}^* : K(x) < k\}| < 2^k.$$

Definition (Binary Entropy Function). Define

$$H_0 = -p\log p - (1-p)\log(1-p).$$

So, $H_0((1/n)\sum_{i=1}^n X_i)$ means $-\bar{X_n}\log\bar{X_n} - (1 - \bar{X_n})\log(1-\bar{X_n})$.

Theorem 7.2.5. *Let x be a binary string. Then,*

$$K(x_1 \dots x_n|n) \leq nH_0(\frac{1}{n}\sum_{i=1}^n x_i) + 2\log n + c.$$

7.3 Kolmogorov Complexity and Entropy

The set of all halting programs forms a prefix free set because no program that leads to a halting computation can be the prefix of another halting program. This allows us to formulate an inequality that is similar to the Kraft inequality for instantaneous codes.

Lemma 7.3.1. *For any computer $\mathcal{U}$,*

$$\sum_{p:\mathcal{U}(p)\ halts} 2^{-l(p)} \leq 1.$$

The next theorem shows the relationship between Kolmogorov complexity and entropy for i.i.d. processes with a finite alphabet.

Theorem 7.3.2. *Let the stochastic process $\{X_i\}$ be drawn i.i.d. according to the probability mass function $f(x)$, $x \in \mathcal{X}$ where $\mathcal{X}$ is a finite alphabet. Let $f(x^n) = \Pi_{i=1}^n f(x_i)$. Then there exists a constant c such that*

$$H(X) \leq \frac{1}{n}\sum_{x^n} f(x^n)K(x^n|n) \leq H(X) + \frac{|\mathcal{X}|\log n}{n} + \frac{c}{n}$$

for all n. Thus,

$$E\frac{1}{n}K(x^n|n) \to H(X)$$

as $n \to \infty$.

7.4 Kolmogorov Complexity of Integers

We can define the Kolmogorov complexity of an integer to be the Kolmogorov complexity of the corresponding binary string.

Definition. The Kolmogorov complexity of an integer n is defined as

$$K(n) = \min_{p:\mathcal{U}(p)=n} l(p).$$

Theorem 7.4.1. *For universal computers $\mathcal{U}$ and $\mathcal{A}$,*

$$K_{\mathcal{U}}(n) \leq K_{\mathcal{A}}(n) + c_{\mathcal{A}}.$$

Since any number can be specified by its binary expansion, we have the following theorem.

Theorem 7.4.2.

$$K(n) \leq \log^* n + c.$$

Theorem 7.4.3. *There are an infinite number of integers n such that $K(n) > \log n$.*

7.5 Algorithmically Random and Incompressible Sequences

Most sequences do not have simple descriptions. If a sequence is drawn at random, it is likely to be complex. The next theorem shows that the probability that a sequence can be compressed by more than k bits is no greater than 2^{-k}.

Theorem 7.5.1. *Let $X_1, \ldots, X_n$ be drawn according to a Bernoulli(1/2) process. Then*

$$P(K(X_1 \ldots X_n|n) < n-k) < 2^{-k}.$$

Thus, most sequences have a complexity close to their length.

Definition (Algorithmically Random). A sequence $x_1, \ldots, x_n$ is *algorithmically random* if

$$K(x_1 \ldots x_n|n) \geq n.$$

Definition (Incompressible). An infinite string x is *incompressible* if

$$\lim_{n\to\infty} K(x_1 \ldots x_n|n)/n = 1.$$

The next theorem is the strong law of large numbers for incompressible sequences.

Theorem 7.5.2. *If a string $x_1x_2\ldots$ is incompressible, then it satisfies the law of large numbers in the sense that*

$$\frac{1}{n}\sum_{i=1}^n x_i \to 1/2.$$

Hence the proportions of 0's and 1's in any incompressible sequence are almost equal.

If a sequence is incompressible, it will satisfy all computable statistical tests for randomness.

The following theorem is the law of large numbers for the Kolmogorov complexity of Bernoulli(θ) sequences which says that the Kolmogorov complexity of a sequence of binary random variables drawn i.i.d. according to a Bernoulli(θ) process is close to the entropy $H_p(\theta)$.

Theorem 7.5.3. *Let $X_1, \ldots, X_n$ be drawn i.i.d. from Bernoulli(θ). Then*

$$\frac{1}{n}K(X_1, \ldots, X_n|n) \to H_0(\theta)$$

in probability.

7.6 Universal Probability

Definition (Universal Probability). The *universal probability* of a string x is

$$P_{\mathcal{U}}(x) = \sum_{p:\mathcal{U}(p)=x} 2^{-l(p)} = \Pr(\mathcal{U}(p) = x)$$

which is the probability that a program randomly drawn as a sequence of fair coin flips $p_1, \ldots$ will print out the string x.

Theorem 7.6.1. *For every computer $\mathcal{A}$,*

$$P_{\mathcal{U}}(x) \geq c'_{\mathcal{A}} P_{\mathcal{A}}(x)$$

for every string $x \in \{0,1\}^*$ where the constant $c'_{\mathcal{A}}$ depends only on $\mathcal{U}$ and $\mathcal{A}$.

7.7 The Halting Problem and the Non-Computability of Kolmogorov Complexity

The halting problem implies the non-computability of Kolmogorov complexity because the only way to find the shortest program is to try all short programs and see which of them work.

7.8 Ω

Definition.

$$\Omega = \sum_{p:\mathcal{U}(p)\text{ halts}} 2^{-l(p)} = \Pr(\mathcal{U}(p)\text{ halts})$$

which is the probability that the given universal computer halts when the input to the computer is a binary string drawn according to a Bernoulli(1/2) process.

Let $\Omega_n = .\,\omega_1 \ldots \omega_n$ denote the first n bits of Ω.

Properties of Ω.

1. Ω is non-computable.

2. Given n bits of Ω, there is an effective procedure to decide the truth of n bit theorems. The basic idea is to run all programs until the sum of the masses $2^{-l(p)}$ contributed by programs that halt equals or exceeds Ω_n. Since $\Omega - \Omega_n < 2^{-n}$, we know that the length of all further contributions to Ω must also be less than 2^{-n}. Hence, no program of length less than n that has not halted will ever halt. We can then decide the halting or non-halting of all programs of length n.

3. Ω is algorithmically random.

Theorem 7.8.1. *Ω cannot be compressed by more than a constant, i.e. there exists a constant c such that*

$$K(\omega_1 \ldots \omega_n) \geq n - c$$

for all n.

7.9 Universal Gambling

7.10 Occam's Razor

7.11 Kolmogorov Complexity and Universal Probability

The next theorem shows the equivalence of $K(x)$ and $\log \frac{1}{p_{\mathcal{U}}(x)}$.

Theorem 7.11.1. *There exists a constant c, independent of x, such that*

$$2^{-K(x)} \leq P_{\mathcal{U}}(x) \leq c2^{-K(x)}$$

for all strings x. Thus, the universal probability of a string x is essentially determined by its Kolmogorov complexity.

The theorem implies that $K(x)$ and $\log \frac{1}{p_{\mathcal{U}}(x)}$ are equivalent universal complexity measures because

$$K(x) - c' \leq \log \frac{1}{p_{\mathcal{U}}(x)} \leq K(x)$$

and complexity defined with respect to two different computers $K_{\mathcal{U}}$ and $K_{\mathcal{U}'}$ are equivalent complexity measures if $|K_{\mathcal{U}} - K_{\mathcal{U}'}|$ is bounded by a constant.

7.12 Kolmogorov Sufficient Statistic

8 Channel Capacity

Definition (Discrete Channel). A *discrete channel* is a system consisting of an input alphabet $\mathcal{X}$ and output alphabet $\mathcal{Y}$, and a probability transition matrix $p(y|x)$ that expresses the probability of observing the output symbol y given that x is sent as input.

Definition (Memoryless Channel). A channel is *memoryless* if the probability distribution of the output depends only on the input at that time and is conditionally independent of previous channel inputs or outputs.

Definition (Information Channel Capacity). The *information channel capacity* of a discrete memoryless channel (DMC) is defined as

$$C = \max_{p(x)} I(X;Y)$$

where the maximum is taken over all possible input distributions $p(x)$.

8.1 Examples of Channel Capacity

1. Noiseless Binary Channel

   Any transmitted bit is received without error. Then, $C = 1$ bit achieved by using $p(x) = (1/2, 1/2)$.

2. Noisy Channel with Non-overlapping Outputs

3. Noisy Typewriter

4. Binary Symmetric Channel (BSC)

   A BSC is a channel where errors occur with probability p. These errors transform a 0 to 1 and vice versa (since the channel is binary). We can bound the mutual information.

$$\begin{aligned} I(X;Y) &= H(Y) - H(Y|X) \\ &= H(Y) - \sum p(x) H(Y|X=x) \\ &= H(Y) - \sum p(x) H(p) \\ &= H(Y) - H(p) \\ &\leq 1 - H(p) \end{aligned}$$

   The information capacity of a BSC with parameter p is $C = 1 - H(p)$.

5. Binary Erasure Channel (BEC)

   BEC is the analog of BSC where bits are lost rather than corrupted. In the BEC, a fraction α of the bits are erased and the receiver knows which bits have been erased. The BEC has 2 inputs (0,1) and 3 outputs (0,1,e).

   The capacity is given by $C = \max_{p(x)}(H(Y) - H(Y|X)) = \max_{p(x)} H(Y) - H(\alpha)$. Considering the event $\{Y = e\}$ and defining $\Pr(X=1) = \pi$, we can compute $H(Y) = H(\alpha) + (1-\alpha)H(\pi)$. Substituting this into the capacity equation, $C = 1 - \alpha$ which is maximized when $\pi = 1/2$.

   Intuitively, this means that since α bits are lost in the channel, we can recover at most $1 - \alpha$ of the bits. It is easy to see how to achieve this bound if there is feedback (simply retransmit lost bits). Surprisingly, it is also possible to achieve this rate without feedback.

8.2 Symmetric Channels

Definition (Symmetric Channel). A channel is *symmetric* if the rows of the channel transition matrix $p(y|x)$ are permutations of each other and the columns are permutations of each other.

Definition (Weakly Symmetric Channel). A channel is *weakly symmetric* if every row of the transition matrix $p(\cdot|x)$ is a permutation of every other row and the column sums are equal.

Theorem 8.2.1. *For a (weakly) symmetric channel,*

$$C = \log|\mathcal{Y}| - H(\mathbf{r})$$

where $\mathbf{r}$ *is a row of the transition matrix and this capacity is achieved by a uniform distribution on the input alphabet.*

The proof of the theorem shows that a uniform input distribution achieves a uniform output distribution.

8.3 Properties of Channel Capacity

1. $C \geq 0$ since $I(X;Y) \geq 0$.
2. $C \leq \log|\mathcal{X}|$ since $C = \max I(X;Y) \leq \max H(X) = \log|\mathcal{X}|$.
3. $C \leq \log|\mathcal{Y}|$ for the same reason.
4. $I(X;Y)$ is a continuous function of $p(x)$.
5. $I(X;Y)$ is a concave function of $p(x)$ which implies that a local maximum is a global maximum.

8.4 Preview of the Channel Coding Theorem

We give some intuition on why C bits of information can be transmitted over a channel.

For each (typical) input n-sequence, there are approximately $2^{nH(Y|X)}$ equally likely Y sequences. We wish to ensure that no two X sequences produces the same Y sequence (otherwise we won't be able to figure out which X sequence was sent).

The total number of possible (typical) Y sequences is approximately $2^{nH(Y)}$. This set is divided into sets of size $2^{nH(Y|X)}$ corresponding to the different input X sequences. The number of disjoint sets is less than or equal to $2^{n(H(Y)-H(Y|X))} = 2^{nI(X;Y)}$. Therefore, we can send at most $2^{nI(X;Y)}$ distinguishable sequences of length n.

8.5 Definitions

Definition (Discrete Channel). A *discrete channel* (denoted by $(\mathcal{X}, p(y|x), \mathcal{Y})$) consists of two finite sets $\mathcal{X}$ and $\mathcal{Y}$ and a collection of probability mass functions $p(y|x)$ (one for each $x \in \mathcal{X}$) such that for every x and y, $p(y|x) \geq 0$ and for every x, $\sum_y p(y|x) = 1$. X is interpreted as the input of the channel and Y is interpreted as the output.

Definition (Discrete Memoryless Channel). The nth extension of the *discrete memoryless channel* (DMC) is the channel $(\mathcal{X}, p(y|x), \mathcal{Y})$ where

$$p(y_k|x^k, y^{k-1}) = p(y_k|x_k)$$

for $k = 1, \ldots, n$.

A channel without feedback is one where the input symbols do not depend on the past output symbols, i.e. $p(x_k|x^{k-1}, y^{k-1} = p(x_k|x^{k-1}$.

If the DMC is used without feedback, then the channel transition function for the nth extension of it reduces to

$$p(y^n|x^n) = \prod_{i=1}^{n} p(y_i|x_i).$$

Definition ((M,n) Code). An *(M,n)* code for the channel $(\mathcal{X}, p(y|x), \mathcal{Y})$ consists of the following:

1. index set $\{1, \ldots, M\}$.
2. encoding function $X^n : \{1, \ldots, M\} \to \mathcal{X}^n$, yielding codewords $X^n(1), \ldots, X^n(M)$. The set of codewords is called the codebook.
3. decoding function $g : \mathcal{Y}^n \to \{1, \ldots, M\}$ which is a deterministic rule that assigns a guess to each possible received vector.

Definition (Probability of Error). The conditional probability of error given that index i was sent is defined as

$$\begin{aligned} \lambda_i &= \Pr(g(Y^n) \neq i | X^n = X^n(i)) \\ &= \sum_{y^n} p(y^n|x^n(i)) I(g(y^n) \neq i) \end{aligned}$$

where I is the indicator function.

Definition (Maximal Probability of Error). The *maximal probability of error* $\lambda^{(n)}$ for an (M,n) code is defined as

$$\lambda^{(n)} = \max_{i \in \{1,\ldots,M\}} \lambda_i.$$

Definition (Average Probability of Error). The *average probability of error* $P_e^{(n)}$ for an (M,n) code is defined as

$$P_e^{(n)} = \frac{1}{M} \sum_{i=1}^{M} \lambda_i.$$

The average probability of error is also the expected probability of error when a (uniformly) random index is chosen on the index set and transmitted through the channel.

Definition (Rate). The *rate* R of a (M,n) code is

$$R = \frac{\log M}{n}$$

bits per transmission.

Definition (Achievable Rate). A rate R is *achievable* if there exists a sequence of $(\lceil 2^{nR} \rceil, n)$ codes scuh that the maximal probability of error $\lambda^{(n)}$ tends to 0 as $n \to \infty$.

Definition (Capacity). The *capacity* of a discrete memoryless channel is the supremum of all achievable rates.

8.6 Jointly Typical Sequences

We decode a channel output Y^n as the ith index if the codeword $X^n(i)$ is jointly typical with the received signal Y^n.

Definition (Jointly Typical Set). The set $A_\epsilon^{(n)}$ of *jointly typical sequences* $\{(x^n, y^n)\}$ with respect to the distribution $p(x,y)$ is the set of n-sequences with empirical entropies ϵ-close ot the true entropies, i.e.

$$\begin{aligned} A_e^{(n)} = \Big\{ &(x^n, y^n) \in \mathcal{X}^n \times \mathcal{Y}^n : \\ &|-\frac{1}{n} \log p(x^n) - H(X)| < \epsilon, \\ &|-\frac{1}{n} \log p(y^n) - H(Y)| < \epsilon, \\ &|-\frac{1}{n} \log p(x^n, y^n) - H(X,Y)| < \epsilon \Big\} \end{aligned}$$

where

$$p(x^n, y^n) = \prod_{i=1}^{n} p(x_i, y_i).$$

Theorem 8.6.1 (Joint AEP). *Let (X^n, Y^n) be sequences of length n drawn i.i.d. according to $p(x^n, y^n) = \prod_{i=1}^{n} p(x_i, y_i)$. Then,*

1. $\Pr((X^n, Y^n) \in A_e^{(n)}) \to 1$ *as* $n \to \infty$.
2. $|A_e^{(n)}| \leq 2^{n(H(X,Y)+\epsilon)}$.
3. *If* $(\tilde{X}^n, \tilde{Y}^n) \sim p(x^n)p(y^n)$ *(i.e.* $\tilde{X}^n$ *and* $\tilde{Y}^n$ *are independent with the same marginals as* $p(x^n, y^n)$*), then*

$$\Pr((\tilde{X}^n, \tilde{Y}^n) \in A_e^{(n)}) \leq 2^{-nI(X;Y)-3\epsilon}.$$

Also, for sufficiently large n,

$$\Pr((\tilde{X}^n, \tilde{Y}^n) \in A_e^{(n)}) \geq (1-\epsilon) 2^{-nI(X;Y)+3\epsilon}.$$

There are $2^{nH(X)}$ typical X sequences and $2^{nH(Y)}$ typical Y sequences but only about $2^{nH(X,Y)}$ jointly typical sequences. So, not all pairs of typical X^n and Y^n are also jointly typical. The probability that any randomly chosen pair is jointly typical is about $2^{-nI(X;Y)}$.

8.7 The Channel Coding Theorem

Theorem 8.7.1 (Channel Coding Theorem). *All rates below capacity C are achievable. For every rate $R < C$, there exists a sequence of $(2^{nR}, n)$ codes with maximum probability of error $\lambda^{(n)} \to 0$. Conversely, any sequence of $(2^{nR}, n)$ codes with $\lambda^{(n)} \to 0$ must have $R \leq C$.*

The code in the proof decodes by joint typicality. To decode, look for a codeword that is jointly typical with the received codeword. With high probability, the transmitted codeword and received sequence are jointly typical because they are probabilistically related. The probability that any other codeword is jointly typical with the received sequence is 2^{-nI}. Therefore, if we have fewer than 2^{nI} codewords, then the probability of error is small.

8.8 Zero-Error Codes

8.9 Fano's Inequality and the Converse to the Coding Theorem

8.10 Equality in the Converse to the Channel Coding Theorem

Examining the conditions for equality in the converse of the coding theorem tells us conditions on the kind of codes that can achieve capacity. These conditions are

1. All the codewords are distinct.
2. The Y_i's are all independent.
3. The distribution of X_i is $p^*(x)$ (the distribution on X that achieves the capacity.

8.11 Hamming Codes

8.12 Feedback Capacity

A channel with feedback is one where all the received symbols are sent back immediately and noiselessly to the transmitter. Surprisingly, the capacity of the channel does not increase with feedback.

8.13 The Joint Source Coding Theorem

The source channel coding theorem tells us that the source code and the channel code can be designed separately and when combined, the combination is able to achieve optimal performance. For example, consider sending speech over a DMC. We can either code the the sequence of speech samples directly into the channel or first compress the speech to its most efficient representation and then use an appropriate channel code to send it over.

This result ties together the data compression theorem (Theorem 3.2.1) and the channel capacity theorem (Theorem 8.7.1).