Information Theory and Related Topics

This summary of selected topics in (and related to) information theory is mostly based on part A of the book *Information, Physics, and Computation* by Marc Mézard and Andrea Montanari (Stanford University, 2009). The chapter about Kolmogorov complexity is based on *Elements of Information Theory* by Cover and Thomas (2006). Other sources are referred to from within the text as needed.

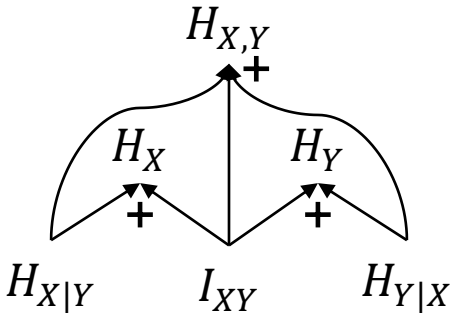
Summarized by Ido Greenberg in 2019.

Contents

Summarizing Table: Intuitive Interpretations

Concept	Interpretation	
	Randomness & Uncertainty	Coding & Information
Surprise $\log 1/p(x)$	Improbability of event	Length of a word
Entropy $H_X(p) \coloneqq E_p[\log 1/p]$	Average surprise (also: log number of effectively-possible states (*AEP*))	Average length of words (also: log number of possible messages)
Cross entropy $H_X(p,q)$	Average surprise in a phenomenon p modeled by q	Length of q-based code used for p-based source
KL-divergence $D(p\|\|q)$ (~difference between p and q)	e^{-nD} is the asymptotic probability of observing p (or farther) instead of q (*Sanov*)	Extra-length of q-based code used for p-based source
Mutual information I_{XY} (~statistical dependence of X and Y)	Randomness of Y after conditioned on X	Possible shortening of Y coding if already knowing X

Relations between entropy (H_X, H_Y), joint entropy (H_{XY}), conditional entropy ($H(X|Y)$,$H(Y|X)$) and mutual information (I_{XY}):



Summary

Introduction

1. ***Entropy*** – measure of randomness or expected surprise: $H_X(p) \coloneqq E[\log_2 1/p(X)]$
   a. More randomness (e.g. richer language) ⇒ longer description per instance of the phenomenon (e.g. longer words) ⇒ each instance of the phenomenon carries more info.
   b. Example – M-faces fair dice: $H = \log_2 M$ = number of bits required to describe a roll.
   c. For M possible values: $\mathbf{0 \le H_X \le \log_2 M}$ – with equalities at determinism ($p(x_0) = 1$) and complete randomness ($p(x) \equiv 1/M$).
   d. $\boldsymbol{H_{X,Y} \le H_X + H_Y}$ with equality iff X,Y are independent.
   e. Behaving nicely under partition of space to $X = X_1 \cup X_2$.
2. Generalizations of entropy to <u>continuous X</u>:
   a. ***Differential entropy*** – $h(f) \coloneqq -\int f \log f \, dx$: useful in spite of some bad properties caused by the non-dimensionless and not-bounded-by-1 $f(x)$ within the log.
   b. ***Limiting density of discrete points***: uses kind of limit of sampling points to normalize f within the log.
3. ***Entropy rate*** of a sequence $\{X_t\}$: $h_X \coloneqq \lim_{t\to\infty} H_{X_t}/t$
4. $H_Y = I_{XY} + H_{Y|X}$ = ***mutual information*** + ***conditional entropy***.
5. ***Cross entropy***: $H_X(p,q) \coloneqq E_p\left(\log\frac{1}{q(x)}\right)$
   a. ***KL-divergence*** – measure of distance between distributions (though not a formal metric): $D(p||q) \coloneqq E_p[\log p/q]$
   b. Length of q-based code used for p-based phenomenon = $H(p,q) = H(p) + D(p||q)$ = inherently required length + extra-length due to approximation of p by q.
6. ***Data processing inequality***: $X \to Y \to Z$ (e.g. $Z = f(Y)$) ⇒ $I_{XZ} \le I_{XY}$.
   a. In particular, any transformation $Z = f(Y)$ can't improve inference from Y about X.
7. **Shannon's source-channel *separation principle***: the problem of information transmission from a single source to a single receiver can be split into 2 independent components – encoding the output of the source (compression) and encoding the transmitted message (error-correction).
8. Data compression:
   a. Language: distribution p over N-long strings $x \in \chi^N$ over an alphabet χ).
   b. **Shannon**: the optimal code's (w: $\chi^N \to \{0,1\}^*$) mean length satisfies $\boldsymbol{H_X \le L_N^* \le H_X + 1}$.
   c. **Huffman code is optimal** over χ^N (for any finite N).
      i. Build decoding tree incrementally from the leaves (strings) by iteratively uniting the 2 least-probable nodes, until only one node remains.
   d. Practical coding is still challenging due to unknown distributions and memory limitations.
9. **Channel coding theorem** (data transmission): given a (possibly noisy) channel $X \to Y$, X can be encoded in advance such that the probability of Y-decoding-error is nearly 0, and the rate of transmitted data ($R = \frac{\text{encoded len}}{\text{original len}}$) is nearly the capacity of the channel ($C = \max_{\text{p(x)}} I_{XY}$).
   a. **I.e. we can encode input as efficiently as in Shannon's theorem, while arranging the distribution of the encoded message $\boldsymbol{p(x)}$ to minimize the info loss in the channel**.
10. ***Fano's inequality***: the decoded message $X \to Y \to \hat{X}$ has error probability $P(\hat{X} \ne X) \ge \frac{H(X|Y)-1}{\log|\chi|}$.

Statistical Physics

11. Physical systems are often modeled by assuming the probability of various states $\{x\}$ to depend on their energy through ***Boltzmann distribution***:

$$\boldsymbol{p_\beta(x) = \frac{1}{Z(\beta)} e^{-\beta E(x)}} \qquad \left(\beta = 1/T,\ Z(\beta) = \Sigma_x e^{-\beta E(x)} = \text{"}\boldsymbol{partition\ function}\text{"}\right)$$

    a. *High-temperature limit* ($\beta \to 0$) ⇒ uniform distribution.
    b. *Low-temperature limit* ($\beta \to \infty$) ⇒ x must be a *ground state* – a global minimum of E.
    c. The ***canonical entropy*** of the system $S(\beta)$ can be calculated directly from the partition function $Z(\beta)$, and counts the (log) possible system states – e.g. $\log|\{\text{all states}\}|$ in the high-temperature limit and $\log|\{\text{ground states}\}|$ in the low-temperature limit.
12. In the ***Thermodynamic limit*** $N \to \infty$, the ***free energy*** $F(\beta) \coloneqq -\frac{1}{\beta}\log Z(\beta)$ is replaced with the ***free energy density*** $f(\beta) \coloneqq \lim_{N\to\infty} F_N(\beta)/N$, whose non-continuities determine ***phase transitions***.
13. ***Ising model***: Model of magnetic materials as cubic lattice with Energy determined by an external magnetic field, and internal interactions between the spins of adjacent molecules.
    a. According to the model, a sufficiently cold material can have strong magnetic properties independently of the magnetic field.

Combinatorial Optimization

14. A ***combinatorial optimization problem*** consists of a finite set of configurations along with cost function (which is often binary – checking some conditions on the configuration).
    a. Continuous optimization (e.g. ***linear programming***) is out of the scope.
15. Possible goals of a combinatorial problem (in decreasing order of difficulty):
    a. ***Optimization***: find the best configuration.
    b. ***Evaluation***: find the best cost.
    c. ***Decision***: is there configuration better than some threshold / satisfying some conditions?
16. ***Polynomial reduction***: converting one problem to another such that the results are preserved.
    a. If B is ***reducible*** to A then B is easier in the sense that solving A results in solving B.
17. Complexity classes:
    a. ***P***: can be decided in polynomial time (e.g. *Euler cycle*).
    b. ***NP***: can be decided in polynomial time be ***non-deterministic Turing machine*** (***NDTM***), which accepts iff any of the configurations is found to satisfy the decision condition.
        i. Equivalently, a configuration can be ***certified*** in polynomial time (e.g. *graph isomorphism*).
    c. ***NP-complete***: NP and also ***NP-hard***, i.e. reducible-to from any other NP problem (e.g. *satisfiability* and *3-satisfiability*, *Hamiltonian cycle*, *traveling salesman*).
    d. ***Co-NP***: its negation is in NP (since acceptance of NDTM is not expressed in a single output true/false, it can't be simply flipped, thus NP & co-NP are not trivially equivalent).
18. It is **often believed yet unknown that $\boldsymbol{P \neq NP}$** (and $NP \neq co-NP$) – otherwise many hard problems (which require exponential time to solve by any known algorithm) must be polynomially-solvable.
19. Other optimization problems worth knowing: *minimum spanning tree*, *max-cut*, *coloring*, *numbers partitioning* (and *zero-sum subset existence*), error correction, energy minimization.

Information and Probability

20. ***Asymptotic equipartition property*** (***AEP***): for iid $X = (X_1 \dots X_n)$, $\frac{1}{n}\log\frac{1}{p(x)} \to H_X(p)$, i.e. we expect to asymptotically observe events only from the $\sim 2^{nH}$ events of probability $\sim 2^{-nH}$.
21. ***Sanov's theorem***: for n iid variables with distribution p, the probability of ***rare events*** – defined by empirical distribution $q \neq p$ (or formally by a set $\{q\}$ of distributions) – decays as $e^{-nD(q||p)}$.
22. ***Maximum entropy distribution***: under constraints $\{E_f[r_i(x)] = \alpha_i\}_{i=1}^m$, the distribution with maximum entropy is of the form $f(x) = e^{\lambda_0 + \sum_{i=1}^m \lambda_i r_i(x)}$.
    a. In particular, Gaussian has maximum entropy under mean & variance constraints.
23. ***Gibbs free energy*** is a convex functional over distributions $G[p] \coloneqq E_p[\text{energy}] + \text{temperature} \cdot H(p)$, from which the Boltzmann distribution can be derived as its minimum, which is useful for understanding the model, calculating rare states probabilities, and finding approximated models.
24. ***Markov chain Monte Carlo*** (***MCMC***) methods construct a Markov chain which is guaranteed to converge to a given distribution p (possibly not normalized), so that by simulation of the Markov chain, a **sampling from p** is applied.
    a. ***Gibbs sampling***: simulate Markov chain of configurations $x^{(t)} \in \chi^d$, where every transition modifies a single dimension $1 \leq i \leq d$ of x according to its marginal distribution $P\left(x_i^{(t+1)} \middle| x_{\hat{\imath}}^{(t)}\right)$.
        i. ***Heat bath algorithm***: Gibbs sampling on Boltzmann distribution wrt some energy function E.
        ii. Example: in a system of d spins, every iteration simulates the change in one spin.
25. **Simulated annealing** performs optimization of a cost function E by applying MCMC sampling on Boltzmann distribution of E with periodically changing $\beta(= 1/T)$ – alternating between low values (allowing exploration through easy transitions) and high values (allowing optimization, since only energy-decreasing transitions remain with positive probability).
    a. Note: applying only the low-temperature limit $\beta \to \infty$ results in ***greedy search*** and is sensitive to local (in terms of the search geometry) minima of E.

Kolmogorov Complexity

26. ***Kolmogorov complexity*** of a string: $\boldsymbol{K(x)} \coloneqq$ the shortest binarily-encoded program that prints x.
    a. Intuition: "how complex it is to describe the string".
    b. Universality: $K(x)$ is invariant to the computer that runs the program (up to a constant).
    c. Upper bound: $\boldsymbol{K(x) \leq l(x) + c}$ (since x can be printed by being hard-coded).
    d. Lower bound: while long strings can have short descriptions (e.g. $x = 0 \dots 0$), counting argument trivially yields $\boldsymbol{|\{x \in \{0,1\}^*: K(x) < k\}| < 2^k}$.
    e. Relations with entropy: for random iid strings $X^n \coloneqq (X_1 \dots X_n)$, $\boldsymbol{E\left[\frac{1}{n}K(X^n)\right] \to H(X)}$.
27. Any ***Incompressible*** infinite sequence (i.e. $\lim_{n\to\infty} \frac{K(x_1 \dots x_n|n)}{n} = 1$) should in particular pass any computable statistical test of randomness.
    a. E.g. a sequence $\{x_i\}$ with $\boldsymbol{\overline{x} \to \theta}$ **can be asymptotically compressed by factor** $\boldsymbol{\approx H(\theta)} = -\theta \log\theta - (1-\theta)\log(1-\theta)$, which is smaller than 1 iff $\theta \neq 1/2$.
28. $K(x)$ is **incomputable** (finding the shortest-encoded program that eventually prints x would require solving the halting problem).
    a. ***Chaitin***'s (incomputable) number: $\boldsymbol{\Omega \coloneqq \sum_{p:U(p)\ halts} 2^{-l(p)} = P(U(p)\ halts)}$.

29. ***Universal probability***: $\boldsymbol{P_U(x)} \coloneqq \sum_{p:U(p)=x} 2^{-l(p)} = P(U(p)=x) \approx 2^{-K(x)}$.
30. ***Minimum description length principle***: $p \coloneqq \underset{\text{p}\in\text{pdfs}}{\operatorname{argmin}} K(p) + \log\frac{1}{p(\{x_1 \dots x_n\})}$.
    a. I.e. given data, the best model is defined to minimize [its description length] + [data description length using corresponding Shannon's code], which in particular follows ***Occam's razor*** principle.
    b. Can be seen as **Bayesian inference with the universal prior** $P_0(p) \coloneqq P_U(p) \approx 2^{-K(p)}$.

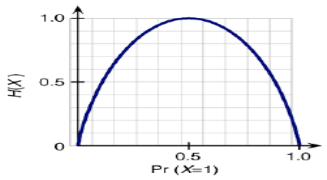
A Sample of Applications in Machine Learning

31. Decision trees construction / entropy minimization.
32. Classification loss function / cross entropy.
33. Feature selection / mutual information minimization.
34. Bayesian networks / mutual information.
35. Independent component analysis (ICA) / mutual information minimization.
36. Variational autoencoders loss function / KL-divergence.

Introduction to Entropy and Information

Entropy

1. ***Entropy*** (discrete): $H_X(p) \coloneqq -\sum_{x\in X} p(x)\log_2 p(x) = \sum p \log 1/p = E[\log 1/p(X)]$
   a. If $\log 1/p$ represents the ***surprise*** of an event, then the entropy is just the **expected surprise**, which is intuitively the **disorder/uncertainty/randomness** in the variable.
   b. The expected surprise cannot be increased by a single "extreme" event:
      i. $\lim_{p\to 0} p\log 1/p = \lim_{p\to 0}\frac{\log 1/p}{1/p} = \lim_{x\to\infty}\frac{\log x}{x} = 0.$
      ii. In particular, **uniform distribution maximizes the entropy** over M events.
      iii. Example – entropy of Bernoulli variable as function of p:



   c. **Units**: ***bits*** (for $\log_2 p$) or ***nats*** (for $\ln p$).
      i. The entropy of **uniform distribution over 2^M events** (e.g. M fair coins) is $\sum 1/2^M \log_2 \frac{1}{1/2^M} = 2^M\left(\frac{1}{2^M}\log_2 2^M\right) = \boldsymbol{M}$ **bits**.
         1. Correspondingly, the entropy of M-faces fair dice is $\log_2 M$.
2. ***KL-divergence*** (***Kullback-Leibler***): $D(q||p) \coloneqq \sum_x q(x)\log\frac{q(x)}{p(x)} = E_q\left[\log\frac{q}{p}\right]$
   a. $D(q||p) \geq 0$ with equality iff $q \equiv p$.
   b. Commonly interpreted as distance between distributions, although not satisfying symmetry or triangle's inequality.
   c. Denoting u :=uniform distribution: $\boldsymbol{D(p||u) = \sum p\log\frac{p}{1/M} = \log_2 M - H(p)}$
      i. Amount of info loss (randomness increase) when approximating p using u.
      ii. In that sense, entropy expresses similarity to the uniform distribution.
   d. KL-divergence is the only measure of difference between distributions which satisfies some properties which are naturally analog to those of entropy (*Arthur Hobson*).
   e. *Fisher information* (see summary in advanced statistical theory) of a parameter in a parametric family of distributions, satisfies $D(p_\theta||p_{\theta+\epsilon}) \approx \epsilon^T I(\theta)\epsilon$ as $\epsilon \to 0$.
3. **Basic properties** of entropy:
   a. $\boldsymbol{0 \leq H_X \leq \log_2 M}$ – reaching the borders at **determinism** ($p(x_0) = 1$) and complete **randomness** ($p(x) \equiv 1/M$).
   b. $\boldsymbol{H_{X,Y} \leq H_X + H_Y}$ with **equality iff X,Y are independent**.
      i. Independent case: $H_{XY} = \sum p_{XY}\log p_{XY} = \sum p_X p_Y(\log p_X + \log p_Y) = H_X + H_Y$.
      ii. I believe that **additivity of independent randomness is what enforces the surprise to be defined as log**. Easier to see for 2 fair dices with M,N faces: we want $\boldsymbol{\text{surprise}(N) + \text{surprise}(M) = \text{surprise}(NM)}$, inducing **surprise $\propto$ log**.
   c. For space partition $X = X_1 \cup X_2$ with $q(X_i) = q_i$ and $r_i(x) \coloneqq P(x|x \in X_i) = p(x)/q_i$:
      $$H_X(p) = H_X(q) + q_1 H_{X_1}(r_1) + q_2 H_{X_2}(R_2)$$
      i. I.e. entropy can be considered separately in subspaces of X.
   d. **Uniqueness**: entropy is the only continuous function satisfying all these properties.

4. Since entropy expresses the (log) amount of different values a variable may accept, and since it is linear in the number of independent variables (e.g. H(2 dices)=2H(dice)), then it can be naturally seen as measure of **information**.
5. ***Entropy rate*** of a sequence $\{X_t\}$: $h_X \coloneqq \lim_{t\to\infty} \frac{H_{X_t}}{t}$
   a. For iid variables it's just $h_X = H(X_1)$.
   b. For stationary process, $h_X = \lim_{t\to\infty} H(X_t|X_1 \dots X_{t-1})$.
   c. Example – random walk on an undirected weighted graph: the stationary distribution is $\pi_i = \frac{\sum_j W_{ij}}{2W}$ (i.e. proportional to the weighted degree of the node), and the entropy rate is $h(\pi) = H(X_2|X_1) = \cdots = H\left(\{W_{ij}/2W\}_{i,j}\right) - H(\{\pi_i/2W\}_i)$, and in particular for n nodes with d (constantly-weighted) edges each, $h(\pi) = \log nd - \log n = \log d$.
   d. In general, to efficiently describe the states of a Markov process, Shannon's code (see below) can be applied on the stationary distribution, resulting in $h(\pi)$ as average length per state.

Differential entropy

1. ***Differential entropy*** – naive generalization of discrete entropy: $\boldsymbol{h_X(f) \coloneqq -\int f(x)\log f(x)\,dx}$.
2. Bad properties:
   a. $\dim f(x) = 1/dx$, thus the argument of the log is not dimensionless.
   b. Not invariant to change of variables.
   c. Can be negative (since $f > 1$ is possible, e.g. $U(0,1/2)$).
3. The conventional alternative generalization of entropy to continuous X is the ***limiting density of discrete points*** (***LDDP***, by *Jaynes*, 1965), which essentially normalizes the argument $f(x)$ of the log by some $m(x)$, that kind of represents the limit density of points in $(x-dx, x+dx)$.
4. Differential entropy still does form a useful global lower bound on expected prediction error:
$$\boldsymbol{E\left[\left(\hat{X}-X\right)^2\right] \geq Var(X) \geq \frac{1}{2\pi e} e^{2h_X(f)}}$$
   a. Note: equality holds above only for estimation of the mean of a gaussian.

Cross entropy and mutual information

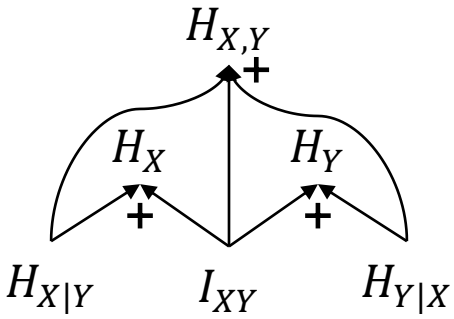
1. ***Cross entropy***: $\boldsymbol{H_X(p,q) \coloneqq E_p\left(\log\frac{1}{q(x)}\right) = -\sum_x p(x)\log q(x)}$
   a. Log-likelihood of a model is its cross entropy with the empirical distribution of the data:

$$\boldsymbol{\log L_X(\theta) = \sum_{i\in samples} \log q_\theta(x_i) = \sum_{y\in values} \#y\cdot\log q_\theta(y) = \sum_y Np_{data}(y)\log q_\theta(y) = NH(p_{data}, q_\theta)}$$

2. ***Conditional entropy***: $\boldsymbol{H_{Y|X} \coloneqq \sum p(x,y)\log p(y|x) = E_X[H_Y|X]}$
   a. "How much information Y adds beyond X": $\boldsymbol{H_{XY} = H_X + H_{Y|X}}$
      i. Independent case: $H_{Y|X} = H_Y \Rightarrow H_{XY} = H_X + H_Y$
3. ***Mutual information*** (*mutual entropy*): $\boldsymbol{I_{XY} \coloneqq \sum p(x,y)\log\frac{p(x,y)}{p(x)p(y)} = E_{XY}\left[\log\frac{p(x,y)}{p(x)p(y)}\right]}$
   a. X-Y symmetric.
   b. **Info of Y = $\boldsymbol{H_Y = I_{XY} + H_{Y|X}}$ = mutual info + info of Y beyond X**.
   c. $I_{XY} = H_Y - H_{Y|X}$ = "how much the randomness of Y reduces when conditioning on X".
   d. "Intersection of information" (in analogy to measures of sets): $I_{XY} = (H_X + H_Y) - H_{XY}$.

e. $X \perp Y \Leftrightarrow I_{XY} = 0$, $X \equiv Y \Rightarrow I_{XY} = H_X = H_Y$.

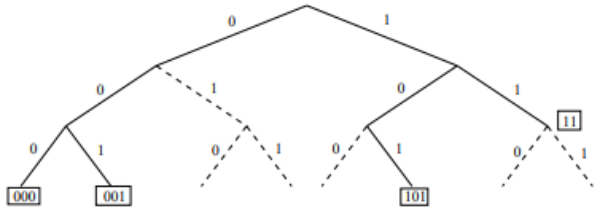
f. A Batman summary of relations:



4. ***Data processing inequality***: If $\boldsymbol{X \to Y \to Z}$, i.e. $p(x,y,z) = p(x)p(y|x)p(z|y)$, **then** $\boldsymbol{I_{XZ} \leq I_{XY}}$.
   a. In particular, if we're interested in X and observe some $Y = Y(X)$, we can't gather more information by any transformation $Z = Z(Y)$.

Data compression

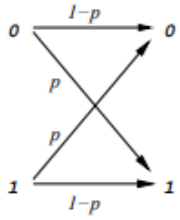
1. *Binary code* over alphabet χ: $w: \chi^N \to \{0,1\}^*$ (each word mapped into a 0-1 sequence)
   a. *Decodable*: one-to-one (i.e. unambiguous).
   b. *Instantaneous*: $w(x_1)$ is never a prefix of $w(x_2)$ (⇒ decodable).
      i. Any instantaneous code can be represented by a binary tree (the word encodes the navigation within the tree, and the leaf is the original string).



2. Average length of code: $L(w) \coloneqq E_{X^N}[l_w] = \sum_{x \in X^N} l_w(x)$
3. **Shannon**: the average length L_N^* of an optimal (i.e. shortest) code satisfies $\boldsymbol{H_X \leq L_N^* \leq H_X + 1}$.
   a. Since the optimal code stores information most efficiently, its $\boldsymbol{L_N^* \approx H_X}$ bits are indeed a reasonable **measure of the inherent information** in an N-chars string in the language.
   b. Proof is based on ***Kraft's inequality*** $\boldsymbol{\sum_x 2^{-l_w(x)} \leq 1}$ for instantaneous codes (proved using tree representation).
   c. The proof is constructive and resulted with ***Shannon code***. However, the construction is not necessarily optimal (i.e. $\boldsymbol{H_X \leq L_N^* < L_N^{Shannon} \leq H_X + 1}$ is possible), and in particular assigns longer words than necessary for very improbable strings.
4. ***Huffman code***: build decoding tree incrementally from the leaves (strings) by iteratively uniting the 2 least-probable available nodes, until only one node remains.
   a. **Huffman code is optimal over $\boldsymbol{\chi^N}$**.
5. Practical challenges in coding:
   a. Applying a general code for N-long strings requires $\boldsymbol{O(|\chi|^N)}$ **memory**.
   b. Finding the optimal code **requires knowledge of the distribution $\boldsymbol{p}$**.

Data transmission

1. *Channel*: defined by $Q(Y|X)$ = Q(output signal | input signal).
   a. In ideal channel, $Y = X$.
   b. *Memoryless channel*: iid, i.e. $Q(Y|X) = \Pi_i Q(Y_i|X_i)$.
      i. Can be visually represented as chart from the input alphabet to the binary output.



2. **Shannon's source-channel *separation principle***: the problem of information transmission from a source through a single transmitter to a single receiver can be split into 2 independent problems:
   a. *Source coding* – compression: encode the output of the source (independently of the channel).
   b. *Channel coding* – noise immunity: encode the information transmitted through the channel (independently of the source).

   In particular, the input of a channel can be assumed to be iid bits without any loss of generality.
   c. Note: in the case of multiple transmitters & receivers, there exist joint source-channel coding schemes which utilize the correlation between the sources for cooperative transmission.
3. ***Capacity*** of channel: $\boldsymbol{C} \coloneqq \max_{\mathbf{p(x)}} \boldsymbol{I_{XY}}$.
   a. The channel distorts the signal X, but the distorted signal Y still carries information about X. The capacity is the amount of this information – the reduction of uncertainty of X caused by knowing Y – under an optimal distribution of X.
   b. Representing **amount of information that can be faithfully transmitted through the channel**.
4. ***Rate*** of code: $R \coloneqq M/N$ = original_length / encoded_length.
   a. ***Redundancy*** of code = $1/R$.
5. ***Block error probability*** of a string $m \in X^M$:

$$\boldsymbol{P_B(m) = P(Y \neq X|m)} = \sum_y Q\big(y\big|x(m)\big) \cdot I(decode(y) \neq m)$$

6. <u>Channel coding theorem</u>: **information can be encoded and faithfully transmitted in rate arbitrarily close to the channel capacity $\boldsymbol{C}$**.
   a. More formally: for any rate $R < C$ – and only for such R – the code w_M (yielding encoded blocks of size M) can "arrange" the distribution of $X = w_M(m)$ to be "nearly optimal", such that $R_M \to R$ and $P_{B,M} \to 0$.
   b. In particular, we can **encode input as efficiently as in Shannon's theorem**, **while making the distribution of the encoded message $\boldsymbol{p(x)}$ minimize the info loss in the channel**.
   c. Proof intuition is based on the requirement #(x's)·#(y's per x) < #(y's) (otherwise 2 x's must be mapped to the same y indistinguishably), where:
      i. #(x's) $\sim 2^M \sim 2^{NR}$.

ii. #(y's-per-x) $\sim 2^{NH_{Y|X}}$ ($H_{Y|X}$ is the info in Y beyond X, i.e. the channel's noise).
iii. #(y's) $\sim 2^{NH_Y}$.

Thus $R < H_Y - H_{Y|X} = I_{XY} \leq C$.

d. Example: in ideal channel $Y \equiv X \Rightarrow I_{XY} = H_X$, thus we can have $\frac{\text{encoded len}}{\text{original len}} \approx H_X$, as in Shannon's theorem for data compression.

7. ***Fano's inequality***: for any estimator $\hat{X}$ yielded by $X \to Y \to \hat{X}$ (e.g. sending X, receiving Y and decoding to $\hat{X}$), the error probability $P_e \coloneqq P\left(\hat{X} \neq X\right)$ satisfies $H(P_e) + P_e \log|\chi| \geq H(X|Y)$.
   a. In particular since $H(P_e) \leq \log_2|\{0,1\}| = 1$, we have $\boldsymbol{P_e \geq \frac{H(X|Y)-1}{\log|\chi|}}$.
8. Cross entropy and KL-divergence in terms of Shannon code:
   a. Shannon code essentially encodes a string m as a word of length $\approx \log 1/p(m)$.
   b. **Cross entropy** $H_X(p,q) = E_p(\log 1/q)$ is thus the **expected <u>length</u> of q-based code used for p-based phenomenon**.
   c. Similarly, $\boldsymbol{D(p||q)} = E_p[\log 1/q - \log 1/p]$ is the **expected <u>extra-length</u> of q-based code used for p-based phenomenon**.
   d. In particular, $\boldsymbol{H(p,q) = H(p) + D(p||q)}$.

Statistical Physics

Intro to thermodynamics

1. Statistical physics deal with probabilistic microscopic modeling of complex physical systems, which results in deterministic macroscopic behavior due to (kind of) law of large numbers.
2. A physical system with N particles can be modeled by *configurations space* χ and *observable functions* of the configurations $O(x)$ (e.g. the *energy* $E(x)$).
    a. An observable satisfies *k-body interaction* if $O(x) = \sum_{i_1 \dots i_k} O_{i_1,\dots,i_k}(x_{i_1}, \dots, x_{i_k})$ (K=2 or 3 for the energy in most models, and K=1 for ideal gas model).
3. The dynamics of a system are often modeled through the energy using ***Boltzmann distribution***:
$$\boldsymbol{p_\beta(x) = \frac{1}{Z(\beta)} e^{-\beta E(x)}} \qquad \left(\beta = 1/T, \qquad Z(\beta) = \Sigma_x e^{-\beta E(x)} = \text{"}\boldsymbol{partition\ function}\text{"}\right)$$
    a. *High-temperature limit* ($\beta \to 0$) ⇒ uniform distribution.
    b. *Low-temperature limit* ($\beta \to \infty$) ⇒ x must be a ***ground state*** – a global minimum of E.
    c. *Boltzmann average* of an observable $O(x)$: $\langle O(x) \rangle \coloneqq E_{p_\beta}[O(x)]$.
4. Example – ***Ising spin*** – a single spin-½ particle ($\chi = \{\pm 1\}$) with energy $E(\sigma \in \chi) \coloneqq -B\sigma$ under magnetic field B: $p_\beta(\sigma) = \frac{e^{\beta B \sigma}}{e^{\beta B} + e^{-\beta B}}$, ***magnetization*** $\coloneqq \langle \boldsymbol{\sigma} \rangle = \frac{e^{\beta B} - e^{-\beta B}}{e^{\beta B} + e^{-\beta B}} = \tanh \beta B$.
5. ***Thermodynamic potentials***: functions of $\beta = 1/T$ and E.
    a. *Free energy*: $F(\beta) \coloneqq -\frac{1}{\beta} \log Z(\beta)$
    b. *Free entropy*: $\Phi(\beta) \coloneqq -\beta F(\beta) = -\log Z(\beta)$
    c. ***Internal energy***: $U(\beta) \coloneqq \frac{\partial}{\partial \beta} \Phi(\beta) = \dots = \langle \boldsymbol{E(x)} \rangle$
    d. ***Canonical entropy***: $S(\beta) \coloneqq \beta^2 \frac{\partial}{\partial \beta} F(\beta) = \dots = -\sum_x p_\beta(x) \log p_\beta(x)$
        i. ⇒ **Shannon entropy of Boltzmann distribution ⇒ counting the** (log) **possible system states** – e.g. $S = \log|\chi|$ for $\beta \to 0$ (high-temp.) and $\log|\{\text{ground states}\}|$ in $\beta \to \infty$ (low-temp.).
        ii. Canonical entropy = $S = \dots = \beta(U + F) = \frac{1}{\mathrm{T}}(\text{Internal Energy} + \text{Free Energy})$.
6. ***Thermodynamic limit***: $N \to \infty$.
    a. *Free energy density*: $f(\beta) \coloneqq \lim_{N \to \infty} F_N(\beta)/N$
    b. *Energy density* u & *entropy density* s are defined analogously.
    c. ***Phase transitions*** occur in non-continuities of $f'(\beta)$ (1st order) or $f''(\beta)$ (2nd order).
    d. ***Energy spectrum***: $N_\Delta(E) \coloneqq$ number of states $x \in \chi$ with $E(x) \in [E, E + \Delta)$.

Ferromagnets, Ising models and spin glasses

1. Magnetic materials contain molecules with magnetic moment, which is a 3D vector tending to align with the magnetic field and with magnetic moments of adjacent molecules.
2. ***Ising model*** (Ernst Ising, 1924): molecules with spins $\sigma \in \{\pm 1\}$ form a 3D cubic grid of size $L \times L \times L$ with energy $\boldsymbol{E}(\{\boldsymbol{\sigma_i}\}) = -\sum_{\boldsymbol{adjacent\ i,j}} \boldsymbol{\sigma_i \sigma_j} - \boldsymbol{B} \sum_{\boldsymbol{i}} \boldsymbol{\sigma_i}$.
3. Temperature limits: $\langle \boldsymbol{\sigma_i} \rangle = \begin{cases} \boldsymbol{\tanh \beta B} & \boldsymbol{\beta/B \to 0} \\ \boldsymbol{\tanh N\beta B} & \boldsymbol{\beta/B \to \infty} \end{cases}$.
    a. **High-temperature limit**: all the states are equally probable (under Boltzmann distribution). However, if we increase B along with $T = 1/\beta$, then each spin still tends to align with B as in the single particle case (see Ising spin above).

   b. **Low-temperature limit**: only the ground state is allowed, which is the state of all spins aligned together (***cooperative response***). This time, if we decrease B along with T, the tendency to align with B remains stronger due to the cooperative response: switching a single spin requires switching the whole system against B.
4. ***Average magnetization***: $M_N(\beta, B) \coloneqq \frac{1}{N}\sum_i \langle\sigma_i\rangle$.
5. ***Spontaneous magnetization*** – large-scale magnetization under tiny magnetic field:
$$M_+(\beta) \coloneqq \lim_{B\to 0^+} \lim_{N\to\infty} M_n(\beta, B)$$
   a. Can be $\neq 0$ only due to the limit $N \to \infty$ (a finite system under $B \to 0$ would have 0 average magnetization).
6. The model is fully solved only for dimensions $d = 1,2$, though many properties of the solution are known also for $d = 3$.
7. In particular, there's a known phase transition $\beta_c < \infty$ in the model for any $d \geq 2$, such that:
   a. Any colder system ($\beta > \beta_c$) has spontaneous magnetization $M_+(\beta) = 1$ (***ferromagnet***) – maximum magnetization under arbitrarily small magnetic field!
   b. Any warmer system ($\beta < \beta_c$) satisfies $M_+(\beta) = 0$ (paramagnet).
8. The book describes the solution for $d = 1$.
9. In summary, from the Ising model we can infer that **a physical system can have strong magnetic properties independently of the surrounding magnetic field, as long as its temperature is low enough**.
10. ***Currie-Weiss model***: each spin interacts with all the other spins (not only adjacent ones).
11. ***Edwards-Anderson model***: $E(\{\sigma_i\}) = -\sum_{adjacent\ i,j} \boldsymbol{J_{ij}}\sigma_i\sigma_j - B\sum_i \sigma_i$.
   a. Capable of modeling materials with ***antiferromagnetic*** interactions (e.g. ***spin glasses***) – where energy of interaction is minimized by <u>opposite</u> alignment of adjacent spins (i.e. $J_{ij} < 0$).
   b. Less understood than the standard Ising model due to the challenge of analysis of a ***frustrated system*** – where the additive components of the energy (namely the energies of the various interactions) cannot be simultaneously minimized, and form a complicated energy landscape.

Introduction to Combinatorial Optimization

1. The opening example of the chapter – ***Minimum Spanning Tree*** – is covered in the Basic Algorithms course summary.
2. ***Combinatorial optimization problem***: set of instances of the problem, each consisting of set of configurations χ and cost function $E: \chi \to R$.
    a. Types of goals:
        i. ***Optimization***: $\underset{x \in \chi}{\operatorname{argmin}} E(x)$.
        ii. ***Evaluation***: $\min_{x \in \chi} E(x)$.
        iii. ***Decision***: for a given E_0, is there any x with $E(x) \leq E_0$?
            1. Equivalently: is there any $x \in L \coloneqq \{x \in \chi | E(x) \leq E_0\}$?
            2. Note: E is often binary is practical problem – "is there a configuration which satisfies some desired conditions?".
    b. An algorithm solves a combinatorial optimization problem if it can get any instance (χ, E) as input and return the solution in finite time.
3. Hardness of problems:
    a. "Hardness" can be compared between problems in terms of **reduction of solution** (i.e. solving one through the solution of the other) and **time complexity** (e.g. existence of algorithm with running time polynomial in the size of an instance, usually assuming that calculating $E(x)$ is polynomial).
    b. Clearly **decision** $\leq$ **evaluation** $\leq$ **optimization** in both senses.
    c. Also **decision** $\geq$ **evaluation** by binary search over the range of costs (up to finite resolution) – just ask whether there's $E(x) < E_0$, and then repeat for higher/lower E_0 according to the answer.
4. Examples of problems:
    a. Find the **ground states of a physical system** with energy E.
        i. In particular, the spin glasses problem can be represented as a ***MAX-CUT problem*** (with the cut being the edges between the positive and the negative spins).
    b. ***Error correction***: decode message such that the average block error probability is minimized.
    c. Given a graph with weights, find a **Minimum Spanning Tree**.
    d. Given a graph, decide whether there's an ***Euler cycle*** (AKA Eulerian circuit – a cycle which passes through each edge exactly once).
        i. Possible algorithm: go over all possible paths from a certain node until a node appears twice or a cycle is completed. If the latter happens – return *true* (exponential time complexity).
        ii. Another possible algorithm: return *true* iff all the nodes have even degree (linear time complexity).
    e. Decide whether there's a ***Hamiltonian cycle*** (visiting each node exactly once) in a graph (NP hard problem).
    f. ***Traveling salesman***; ***assignment*** (matching pairs between two groups); ***satisfiability***; ***coloring*** of graphs; ***numbers partitioning*** to two sum-equal subsets; etc…
5. ***Polynomial reduction***:

   a. A problem B is ***polynomially reducible*** to A (meaning "not harder") if there exists a mapping $R: B \to A$ from instances of B to instances of A such that:
      i. For any decision instance $I \in B$, $R(I) = yes \Leftrightarrow I = yes$.
      ii. $R(I)$ is computable in polynomial time in $|I|$.
      iii. $|R(I)|$ is polynomial in $|I|$.
   b. Example: the Hamiltonian cycle problem is reducible to the satisfiability problem.
6. Complexity classes:
   a. Polynomial (***P***): problems for which there exists algorithm with polynomial running time.
   b. Non-deterministic polynomial (***NP***): problems for which there exists a non-deterministic algorithm – an algorithm that can run in polynomial time on ***non-deterministic Turing machine*** (***NDTM*** – a machine that commits distributional computations and "accepts" iff any of its distributional branches accepts).
      i. Equivalently, a problem is in NP iff a suggested solution can be verified ("is $x \in L$?") in polynomial time (***short certification***).
   c. ***NP-complete***: problems in NP which are also ***NP-hard***, i.e. any other problem in NP is polynomially reducible to them.
   d. ***Co-NP***: decision problems whose complementarians are in NP (i.e. "does $\forall x: x \in L^c$?").
      i. The difference from NP is derived from the **asymmetric definition of NDTM** to accept iff *any* of its branches returns "yes": it can't be configured to accept iff *all* its branches return "no".
      ii. In particular, note that the acceptance of the NDTM is quite abstract, in the sense that it cannot be used to simply reverse the answer (i.e. we can't send it to a *not* gate which just returns "yes" if the machine rejected and "no" if it accepted).
   e. **Examples**:
      i. **NPC problems**: satisfiability (*Cook, 1971*), 3-satisfiability (satisfiability with clauses of length 3), Hamiltonian cycle, traveling salesman.
      ii. Problems in NP with **unknown precise classification** (quite rare – most of the known problems have known classification…): graph isomorphism.
      iii. Problems which are **not in NP**: some non-decision problems, e.g. the optimization & evaluation variants of the traveling salesman.
      iv. Co-NP problems: any complementary of NP problem – e.g. "is there no partition to 2 equal-sum subsets?" or equivalently "are all partitions to 2 subsets result in different sums?".
   f. Clearly $\boldsymbol{P \subseteq NP}$ and $\boldsymbol{NPC \subseteq NP}$. It is **unknown whether $\boldsymbol{P = NP}$ and $\boldsymbol{NP = co - NP}$**.
      i. It is actually often believed that there's no polynomial algorithm for NP-hard problems, i.e. that $P \neq NP$. In that case P and NPC are disjoint, and it was also proved that there exist problems in NP which are neither in P nor NPC.
7. Continuous optimization problems are naturally out of the scope of combinatorial optimization. ***Linear programming***, which deals with optimization problems of linear cost under linear constraints, is a common example and is briefly covered in the Basic Algorithms course summary.

Information and Probability

Asymptotic equipartition, Sanov's theorem and maximum entropy

1. ***Asymptotic equipartition property*** (***AEP***): for iid $X_1 \dots X_n$, $\boldsymbol{\frac{1}{n}\log\frac{1}{p(X_1 \dots X_n)} \to H_X(p)}$ in probability.
   a. Proved directly by LLN: $-\frac{1}{n}\log p(X_1 \dots X_n) = -\frac{1}{n}\sum \log p(X_i) \to -E[\log p(X)] = H_X(p)$.
   b. The formulation of the theorem looks at probability as a statistic – a function of the data – and refers to its asymptotic value. The interpretation of the consequences, however, looks at probability as we're used to – "which data is likely to occur" – which in this case is **all the $\boldsymbol{2^{nH}}$ events with probability nearly $\boldsymbol{2^{-nH}}$** (in particular nearly equal probability).
   c. In particular, for any ***typical set*** $\boldsymbol{A_\epsilon^{(n)}} \coloneqq \left\{(x_1 \dots x_n) \in \chi^n \mid 2^{-n(H(p)+\epsilon)} \le p(x_1 \dots x_n) \le 2^{-n(H(p)-\epsilon)}\right\}$ and large enough $n \in N$, we have $\boldsymbol{P\left(A_\epsilon^{(n)}\right) > 1-\epsilon}$ and $\boldsymbol{\left|A_\epsilon^{(n)}\right| \approx 2^{nH(p)}}$.
   d. Example: in $n = 10^6$ tosses of an unfair coin $\boldsymbol{p = 0.9}$**, we expect to have nearly $\boldsymbol{9 \cdot 10^5}$ heads, although the single event $\boldsymbol{X = x_1 \coloneqq (1)_{i=1}^{10^6}}$ actually has larger probability $\boldsymbol{p(x_1) > 2^{-nH}}$**. This is possible since although its probability as a single event remains the largest, it remains a single event among the exploding-2^n possible events.
2. ***Sanov's theorem***: if $\{X_i\}_i \sim P$ are iid with empirical distribution $\hat{P}_n \coloneqq \frac{1}{n}\sum_i \delta_{X_i}$, and Γ is a "nice" set of distributions, then $\lim_{n\to\infty}\frac{1}{n}\log P(\hat{p}_n \in \Gamma) = -\inf_{q\in\Gamma} D(q||p)$.
   a. "Nice" set of distributions here means $\inf_{q\in\Gamma^\circ} D(q||p) = \inf_{q\in\bar{\Gamma}} D(q||p)$.
   b. The theorem essentially describes the **decay rate of probability of *rare events*** by $\boldsymbol{P(\hat{p}_n \in \Gamma) \approx \inf_{q\in\Gamma} e^{-nD(q||p)}}$ (for large n, with D in units of nats).
      i. In particular, if $p \in \Gamma$ we just have $P(\hat{p}_n \in \Gamma) \to 1$, which is kind of LLN.
   c. Note: the "rarity" of a set of events is asymptotically determined only by the least rare event in the set, corresponding to the empirical distribution q^*. In particular, the ***conditional limit theorem*** states that $p(X_1|\hat{p}_n \in \Gamma) \to q^*$ in probability.
   d. Example: if $\boldsymbol{X_i \sim N(0,1)}$ and $\boldsymbol{\Gamma = \left\{q \mid E_q[X_i] \ge t\right\}}$ $(t > 0)$, then $P(\hat{p}_n \in \Gamma)$ is the probability that the empirical distribution will be in Γ, i.e. the probability that the empirical mean will deviate from 0 by t ($\langle x\rangle \ge t$), i.e. $\boldsymbol{P(\hat{p}_n \in \Gamma) = \int_{t\sqrt{n}}^{\infty}\frac{e^{-x^2/2}}{\sqrt{2n}}dx \approx e^{-nt^2/2}}$ (I didn't validate the calculation).
   e. It is claimed that Sanov's theorem and other important results **can be generalized to continuous spaces χ** by using ***fields theory***, and in particular the ***saddle point method***.
3. ***Maximum entropy distribution***: the distribution f which **maximizes the (possibly differential) entropy $\boldsymbol{h(f)}$ under the constraints** $\left\{E_f[r_i(x)] = \alpha_i\right\}_{i=1}^m$, is of the form $\boldsymbol{f(x) = e^{\lambda_0 + \sum_{i=1}^m \lambda_i r_i(x)}}$.
   a. Note: in lack of constraints, the entropy is maximized by the uniform distribution.
   b. Example – $E[X] = 0$ and $E[X^2] = \sigma^2$: the entropy is maximized by $f(x) = e^{\lambda_0 + \lambda_1 x + \lambda_2 x^2}$ with corresponding constants. Note that f is of the form of a Gaussian, thus necessarily $f \sim N(0,\sigma^2)$ – i.e. **Gaussian has the maximum entropy given mean and variance**.
      i. This can represent, for example, to the distribution of 1D velocities of molecules of standing gas with certain temperature.

Gibbs free energy

1. The Boltzmann distribution of a physical system was previously written as $p_\beta(x) = \frac{1}{Z(\beta)} e^{-\beta E(x)} = e^{-\beta E(x) - F(\beta)}$ (normalized in terms of the partition function and the free energy, respectively).
2. Gibbs free energy is a real functional over space of distributions:
$$\boldsymbol{G[p] \coloneqq E_p[E(x)] + T \cdot H(p)} = \sum_x p(x)E(x) + \frac{1}{\beta}\sum_x p(x)\log p(x) = \frac{1}{\beta}\boldsymbol{D\left(p||p_\beta\right)} + F(\beta)$$
   a. Note: p is the true distribution and p_β is the Boltzmann distribution.
3. Clearly, the Gibbs free energy $G[p]$ is **convex**, $\boldsymbol{p \coloneqq p_\beta}$ **minimizes** it, and $G\left[p_\beta\right] = F(\beta)$.
4. The Gibbs free energy is thus a concept from which the **Boltzmann distribution can be derived (as minimization of some energy functional)**, and in particular it allows derivation of convenient approximated probability models for various systems.
5. In particular, by writing Gibbs free energy in terms of $D\left(p||p_\beta\right)$, Sanov's theorem implies that given N many physical systems, the probability of "atypical" empirical distribution $p \neq p_\beta$ is exponentially small: $P(p) \sim e^{-ND(p||p_\beta)} = e^{-N\beta(G[p]-F(\beta))}$.

Markov Chain Monte Carlo sampling

1. Why sampling of a configuration among N many configurations may be challenging?
   a. Too many possible configurations.
   b. Typical (in terms of probability) configurations may be exponentially rare (in terms of simple counting).
   c. The probability may be known only up to a constant (e.g. Boltzmann distribution with unknown partition function).
2. ***Markov chain*** (MC) is a random process $\{X_i\}$ where $P(X_{i+1})$ depends only on X_i.
   a. MC is defined by its ***transition matrix*** $T_{mn} \coloneqq P(X_{i+1} = x_n | X_i = x_m)$.
   b. MC is ***irreducible*** iff any state is reachable from any state with positive probability (i.e. $\forall x, y, \exists k$: $P(X_{i+k} = y | X_i = x) > 0$).
   c. MC is ***aperiodic*** iff reaching y from x is possible within any number of steps $k \geq k_0$ (rather than, for example, only in multiplications of some period, as in the simple chain $x \leftrightarrow y$ with period 2).
   d. $\pi(x)$ is ***stationary distribution*** (or steady-state) iff $T\pi = \pi$ (i.e. $\forall y$: $\pi(y) = \sum_x \pi(x) T_{xy}$).
      i. *Detailed balance condition* – sufficient for stationarity: $\pi(x)T_{xy} = \pi(y)T_{yx}$.
   e. **Any irreducible aperiodic Markov chain satisfies** $\boldsymbol{\lim_{t\to\infty} P(X_t = x) = \pi(x)}$ **and** $\boldsymbol{\langle f(x) \rangle_{\{X_1 \dots X_t\}} \to E_\pi[f]}$ (for any f, almost surely).
3. ***Markov chain Monte Carlo*** (***MCMC***) methods **construct MC which is guaranteed to converge to a given distribution $\boldsymbol{\pi}$** (**possibly not normalized**), in order to **apply sampling** from the distribution.
4. ***Gibbs sampling*** is one such method which essentially **simulates dimensionally-local transitions in multi-dimensional configuration space $\boldsymbol{\chi^d}$** (e.g. the spins of d molecules): draw a random $x^{(0)} \in \chi^d$ uniformly; then iteratively over t, choose a random dimension $1 \leq i \leq d$, and draw a new value for $x_i^{(t)} \to x_i^{(t+1)}$ according to the marginal distribution $P\left(X_i^{(t+1)} | X_{\hat{i}}^{(t)} = x_{\hat{i}}^{(t)}\right)$ (note that only the marginal distribution needs to be normalized).
   a. ***Heat bath algorithm***: Gibbs sampling wrt Boltzmann distribution (e.g. as in the d spins example).

   b. The **locality** in Gibbs sampling, which determines the changes considered every iteration, **can be generalized** from "identical up to a single coordinate" (as above) to anything **defined by a connected graph**.

Simulated annealing

1. **Any optimization problem can be represented as a statistical mechanics problem**: by interpreting the cost of configurations $E(x)$ as energy, **assuming Boltzmann distribution and applying the low-temperature limit**, the probability of the system is guaranteed to concentrate around the ground states $\mathrm{argmin}_{\mathrm{x}} E(x)$.
2. Since **MCMC can sample from Boltzmann distribution**, one may try to **use it for optimization**.
3. By assigning Boltzmann distribution $\pi(x) \propto e^{-\beta E(x)}$ in the stationarity equation $\pi(y) = \sum_x \pi(x) T_{xy}$ we have $1 = \sum_x e^{-\beta(E(x)-E(y))} T_{xy}$, hence $\forall x, y\colon 0 \le T_{xy} \le e^{-\beta(E(y)-E(x))}$, which goes to 0 for any $E(x) < E(y)$ in the low-temperature limit.
   a. In other words, the **low-temperature limit enforces the energy to only gets lower** (**through zeroed transition probabilities**), which prevents the Markov chain from being irreducible and removes the guarantee to converge to the equilibrium distribution.
   b. In particular, since the practical simulated process can only consider local transitions, it becomes a ***greedy search*** and thus is **sensitive to local minima of $\boldsymbol{E}$**.
      i. Note: "local" is in terms of the geometry of the search, e.g. "identical up to one coordinate" in the case of the heat bath algorithm above.
4. A suggested solution is to apply MCMC on Boltzmann distribution in non-zero temperature ($\beta < \infty$), which should reach the ground state x_0 within expected time $E[T] = 1/p_\beta(x_0)$.
5. Since often $\lim_{N\to\infty} p_\beta(x_0) = 0$, β must be scaled with N to reach x_0 within finite time, but then we reach the low-temperature limit again. **To compromise between the need to avoid searching over the whole exponentially many sates but rather prefer low-energy states; and the need to avoid degenerated transitions and sensitivity to local minima – an annealing schedule is used**, where β accepts values alternately low (for exploration) and high (for optimization).
   a. Note: the resulted Markov chain is time-dependent (rather than *homogeneous*).
6. The terminology ***simulated annealing*** comes from the process of shaping steel by alternately heating and cooling it.

Kolmogorov Complexity

This section is based on chapter 14 in the 2nd edition of *Element of Information Theory* by Cover & Thomas.

Definition, universality and basic properties

1. ***Universal computer***: a machine equivalent to Turing machine, in the sense that it can both implement and be implemented by a Turing machine.
2. ***Kolmogorov complexity*** of a string x given a universal computer U: $\boldsymbol{K_U(x) \coloneqq \min_{p:U(p)=x} l(p)}$
   a. I.e. the **shortest binarily-encoded program whose output is $\boldsymbol{x}$**.
   b. Intuitively: **"how complex it is to describe the string"**.
      i. In particular, if x can be constructively described in free language within n 8-bits characters, then its Kolmogorov complexity can't be more than $8n$ bits.
   c. Terminology: Kolmogorov complexity = *algorithmic complexity* = *descriptive complexity*.
3. ***Berry paradox***: **"the shortest number not nameable in under 10 words"**. This illustrates the problem with not well-defined meaning of "nameable" (or "describable"). In Kolmogorov complexity we use the meaning "can be programmatically described", i.e. "can be described for printing by a computer".
4. Conditioning on knowledge of the string length $l(x)$:
   a. It is unclear in the definition of $K_U(x)$ whether the program p can assume to know $l(x)$.
   b. It's actually doesn't matter a lot, since the $l(x)$ can always be encoded in the beginning of p in $O(\log l(x))$ bits (note: if x is an integer then $\log l(x) = \log\log x$).
      i. Example of such a code: write the binary representation of $l(x)$ with duplicated digits (i.e. $0 \to 00,\ 1 \to 11$), and mark the end of the encoded length with 01.
   c. When it does matter, we'll denote by $\boldsymbol{K_U(x|n)}$ the complexity conditioned on $l(x)$.
5. ***Universality*** of K_U: for any universal computers A, B, $\boldsymbol{K_B(x) \le K_A(x) + C_{AB}}$.
   a. I.e. Kolmogorov complexity is independent of the specific computer (up to a constant).
   b. Proof: just use C_{AB} bits to encode implementation of A in B.
   c. It is actually claimed that for universal machines, $C_{AB} \equiv C$, accordingly re-denoting $K(x) = K_U(x)$. Maybe I don't understand how exactly a universal computer is defined.
6. Bounds on $K(x)$:
   a. **Upper bound:** $\boldsymbol{K(x) \le l(x) + c}$ (proof: just print x hard-coded).
      i. Note: the constants (both in the upper bound and in the conversion between universal computers) may be very large, so the whole theory is mainly relevant for either huge strings or conceptual purposes.
      ii. Note: $2\log l(x)$ may be added to the upper bound if it's not conditional on the length $l(x)$ (see "conditioning on knowledge of…" above).
   b. **Lower bound:**
      i. $\boldsymbol{|\{x \in \{0,1\}^*: K(x) < k\}| < 2^k}$ (at most 2^k strings can be encoded in k bits…).
      ii. For iid $X_1 \dots X_n \sim Bernoulli(1/2)$, $\boldsymbol{P(K(X_1 \dots X_n|n) < n-k) < 2^{-k}}$.
         1. I.e. large compression is rare.
7. ***Algorithmically random*** sequence: $K(x_1 \dots x_n|n) \ge n$.
   a. Note: by counting argument, for any n there exists at least one such sequence.
8. ***Incompressible*** infinite sequence: $\lim_{n\to\infty} \frac{K(x_1 \dots x_n|n)}{n} = 1$.

   a. For any incompressible sequence, $\frac{1}{n}\sum_{i=1}^{n} x_i \to \frac{1}{2}$ ("***strong law of large numbers***").
      i. I.e. any incompressible sequence has similar proportion of 0s and 1s.
   b. In general, incompressible sequence satisfies all the computable statistical tests for randomness, otherwise the statistical structure could be exploited to describe it shortly.
9. Examples:
   a. Sequence of 01s: $K(0101\dots01|n) = c$
   b. Digits of π: $K(\pi_1\dots\pi_n|n) = c$
   c. An integer $n \in N$: $K\big(n|l(n)\big) \leq \log n + c$ (just hard-code the binary repr.)
   d. Imbalanced binary sequence: $K(x_1\dots x_n|n) \leq nH_0 + \frac{1}{2}\log n + c$
      i. $H_0 \coloneqq -\bar{x}\log\bar{x} - (1-\bar{x})\log(1-\bar{x})$ $\quad(\bar{x} \coloneqq \sum x_i/n\,)$
      ii. The proof is based on the constructive idea "generate all sequences with $k = \sum x_i$ ones, then peak the i^{th} sequence".
      iii. For $\bar{x} = 1/2$ we have $H_0 = 1$, yielding the global upper bound $O\big(l(x)\big) = O(n)$.
      iv. The bound gets closer to 0 as $\bar{x}$ gets more distant from $1/2$.
10. **Kolmogorov complexity and entropy**:
   a. $\boldsymbol{E\left[\frac{1}{n}K(X^n)\right] \to H(X)}$ for iid $\{X_i\}$ above finite alphabet.
   b. $\boldsymbol{\frac{1}{n}K(X_1\dots X_n|n) \to -\theta\log\theta - (1-\theta)\log(1-\theta)}$ in probability for iid $Bernoulli(\theta)$.
      i. Note: the two properties are taken from two different subsections, and it is unclear whether the differences in setup (finite alphabet vs. Bernoulli) and notation (X^n vs. $X_1\dots X_n$) are meaningful.

Computability of Kolmogorov complexity and Chaitin's number

1. Example (non-triviality of Kolmogorov complexity evaluation): the binary expansion of $\sqrt{2}-1$ for $n = 100$ bits, for example, would pass most conventional tests for randomness, even though it can be described very shortly.
2. **Kolmogorov complexity of a string x is not computable**, since computing it (or even validating it) would require going over all programs (or all shorter programs for validation) and tell whether they return x or not – which would in particular find out whether they stop or not, which **would solve the halting problem**.
   a. Note: **unlike finding the shortest-running-time program**, to find the **shortest-encoded program** (which may have longer running time than other valid programs), we can't just go over many programs in parallel until any of them successfully halts.
3. ***Chaitin's number***: $\boldsymbol{\Omega \coloneqq \sum_{p:U(p)\ halts} 2^{-l(p)}}$
   a. Note: the encoded programs that halt are a prefix-free set (none of them is a prefix of any other), hence they satisfy Kraft inequality, thus $0 \leq \Omega \leq 1$.
   b. By drawing a random program by sequentially drawing its bits as $Bernoulli(0.5)$ until a valid program is yielded, we have $\boldsymbol{\Omega = P(U(p)\ halts)}$.
   c. **Ω is non-computable**, since computing it would require solving the halting problem.
   d. It can be shown that knowing Ω (in resolution of n bits) **would allow us to decide the truth of any provable mathematical theorem** (phraseable in less than n bits).
      i. Example: by coding a program that goes over all integer quartettes $\{(n,a,b,c) \in N^4 | n \geq 3\}$ until $a^n + b^n = c^n$, and testing whether it halts or not, the truth of ***Fermat's last theorem*** can be decided.

e. Ω is algorithmically random, in the sense that its binary representation up to n bits can't be compressed by more than a constant ($\exists c, \forall n: K(\omega_1 \dots \omega_n) \geq n - c$).

Universal probability, Occam's razor and minimum description length principle

1. ***Universal probability*** of a string: $\boldsymbol{P_U(x) \coloneqq \sum_{p:U(p)=x} 2^{-l(p)}}$
   a. $\boldsymbol{P_U(x) = P(U(p) = x)}$, i.e. the universal probability is **the probability that a program $\boldsymbol{p}$ which is randomly drawn as a sequence of Bernoulli(1/2) bits will print $\boldsymbol{x}$**.
   b. The universal probability is independent of the computer up to a constant multiplication.
2. Kolmogorov complexity and universal probability: $\boldsymbol{P_U(x) \approx 2^{-K(x)}}$.
   a. More precisely $2^{-K(x)} \leq P_U(x) \leq c2^{-K(x)}$ (or $K(x) - c \leq \log \frac{1}{P_U(x)} \leq K(x)$).
   b. Note: $\boldsymbol{\log 1/p_U(x) \approx K(x)}$ **is somewhat analog to** $\boldsymbol{E[\log 1/p] = H(X)}$.
   c. In particular, for a standard textual string x, the probability $\approx 2^{-K(x)}$ of a randomly drawn program to print x is significantly larger than the probability $2^{-l(x)}$ of a randomly drawn string to be x.
      i. I.e. we should sit the monkey in front of a terminal, not a word processor.
3. ***Occam's razor*** principle: among explanations consistent with the data, the simplest should be chosen.
   a. Example: general relativity is simpler than a [Newtonian theory patched to fit the 20th century observations].
   b. Kolmogorov complexity gives a natural measure of simplicity.
   c. Note: **choosing explanation with small Kolmogorov complexity is similar to Bayesian approach with the universal probability as a prior**.
4. Example: will the sun rise tomorrow, given that it has risen in all n days of known history?
   a. The universal probability of a sequence beginning with $n+1$ 1s is $\sum_y p(1^n 1y) \approx p(1^\infty) = c > 0$. Similarly, $\sum_y p(1^n 0y) \approx p(1^n 0) \approx 2^{-\log n} = 1/n$ (since the program must describe n, which in general takes $\sim \log n$ bits), hence the probability that the sun won't rise tomorrow is $\approx \frac{1/n}{c+1/n} \approx 1/cn$.
   b. Laplace got similar result with Bayesian approach, assuming the rising of the sun is $Bernoulli(\theta)$ with a uniform prior: $P(X_{n+1}|X_1 = 1 \dots X_n = 1) = \frac{P(X_1=1 \dots X_{n+1}=1)}{P(X_1=1 \dots X_n=1)} = \frac{\int_0^1 \theta^{n+1} d\theta}{\int_0^1 \theta^n d\theta} = \frac{n+1}{n+2}$, i.e. the probability that the sun won't rise tomorrow is again $\approx 1/n$.
5. <u>Minimum description length principle</u>:
   a. Goal: fit data with a probability model.
   b. Empirical distribution: best possible fit (and a brute overfit): $f(x) = \sum_{i=1}^n \frac{1}{n} \delta(x_i)$.
      i. ***Kernel density estimation***: smoothing of the empirical fit.
   c. ***Maximum likelihood*** among a parametric family of distributions: $\operatorname{argmax}_\theta p_\theta(\{x_1 \dots x_n\})$.
   d. **Kolmogorov-complexity-based fit**: $\boldsymbol{p \coloneqq \operatorname{argmin}_{p \in pdfs} K(p) + \log \frac{1}{p(\{x_1 \dots x_n\})}}$
      i. It's kind of **Bayesian inference with the universal prior** $P(p) \coloneqq P_U(p) \approx 2^{-K(p)}$.
      ii. It also follows **Occam's Razor** through the ***minimum description length principle*** – it's the shortest way to sequentially describe the distribution p and then the data $\{x_1 \dots x_n\}$ using p-based Shannon's code.

A Sample of Applications of Information Theory in Machine Learning

1. This section is mostly based on this must-know information theory concepts in AI.
2. **Decision trees construction**: splitting nodes are often chosen by the criterion of minimum post-split **entropy** – since the data is wished to be as homogeneous (non-random) as possible after the split.
3. **Cross entropy** is a common **loss function** in classification problems: the classifier (e.g. a neural network) returns $0 \leq y \leq 1$ which is interpreted as $y = P(x \in Class_1)$, and the loss is $L = \begin{cases} \log\frac{1}{y} & x \in C_1 \\ \log\left(\frac{1}{1-y}\right) & x \notin C_1 \end{cases}$.
4. **Feature selection**: independent features can increase the information-per-feature (or per degree of freedom, or per model complexity), which allows better exploitation of the information (and in particular reduces overfitting). While correlation is only capable of capturing linear relationship, **mutual information** can truly guarantee small dependence between different features.
5. The inferential connections between variables in **Bayesian networks** are often based on their **mutual information**.
6. **Independent component analysis** (**ICA**) attempts to decompose a signal into a sum of as independent components as possible, often through minimization of their **mutual information**.
7. ***Variational autoencoders*** (***VAE***) are a variant of autoencoders, which is intended to find efficient (usually in terms of compression) representation of data by learning to predict the data (as output) from itself (as input), where the architecture of the model (usually neural network) enforces passage through a restricted (usually just smaller) layer of memory: $X \to Y_{small} \to \hat{X}$. Variational autoencoders use **KL-divergence** as (a main component in) their loss function.
8. This source briefly describes further applications such as bottleneck research in deep neural networks and rare events prediction.