INTRODUCTION TO GRAPH ENTROPY

VIKTOR HARANGI

Abstract. This is an introduction to the notion of (conditional) graph entropy. It is mainly based on [3]. It was created as supplementary material to the website
`https://www.renyi.hu/~harangi/ge.htm`.

1. Graph entropy: a visual introduction

1.1. **A coding problem with indistinguishable letters.** Suppose that the discrete random variable X takes values in a finite alphabet $\mathcal{X}$. If we take a large number of IID copies $X_1, \dots, X_\ell$, then with high probability the outcome will be a so-called *typical sequence*, in which the frequency of each letter $x \in \mathcal{X}$ is close to its probability

$$p_x := \mathbb{P}(X = x).$$

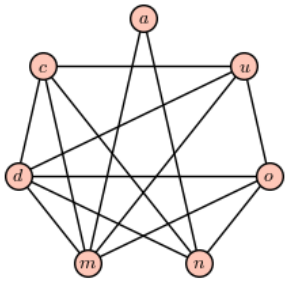
Recall that the *Shannon entropy* of X is defined as

$$H(X) = \sum_x -p_x \log p_x.$$

Loosely speaking, any typical sequence has probability roughly $\exp\big(-\ell H(X)\big)$, and hence the number of typical sequences is roughly $\exp\big(\ell H(X)\big)$. This observation essentially proves the following classical result: $H(X)$ gives the minimal *code rate* required to encode the IID sequence (X_i) such that, with high probability, it is uniquely decodable.

Körner introduced [5] a natural variant of this coding problem, where not every pair of letters can be distinguished. Let G be a graph with vertex set $V(G) = \mathcal{X}$ describing which pairs are distinguishable: $x, x' \in \mathcal{X}$ can be distinguished if and only if xx' is an edge of G. Furthermore, we say that the sequences $x_1, \dots, x_\ell$ and $x'_1, \dots, x'_\ell$ are distinguishable if x_i and x'_i are distinguishable for at least one index i. In this variant we wish to encode the IID sequence with high probability in a way that distinguishable sequences are mapped to different codewords. Again, we are interested in the minimal achievable code rate that we call the *graph entropy* of X w.r.t. G and denote by $H_G(X)$.

For example, consider the alphabet $\mathcal{X} = \{a, c, d, m, n, o, u\}$ with the graph G below:



That is, the following pairs of letters are indistinguishable:

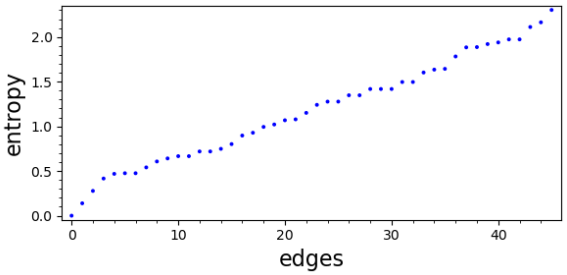
a	a	a	a	c	m	n
c	d	o	u	o	n	u

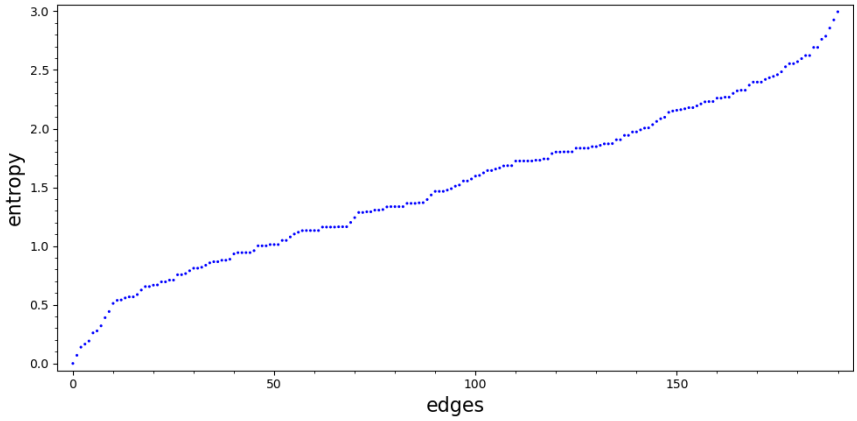
Some examples for distinguishable and indistinguishable words (sequences):

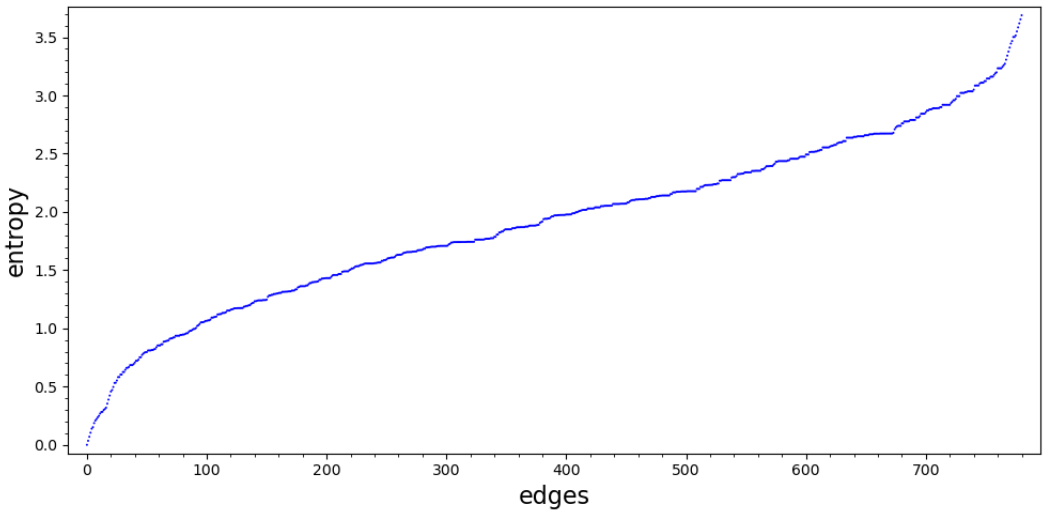
indistinguishable pairs			distinguishable pairs		
moana	*uncoad*	*mad*	*moana*	*uncoad*	*dad*
nodua	*nuocca*	*mud*	*nodun*	*duonca*	*mud*

1.2. **From empty to complete.** For the complete graph G, we get back the original coding problem, so $H_G(X) = H(X)$, while for the empty graph any two sequences are indistinguishable, and hence $H_G(X) = 0$. In general, $0 \leq H_G(X) \leq H(X)$.

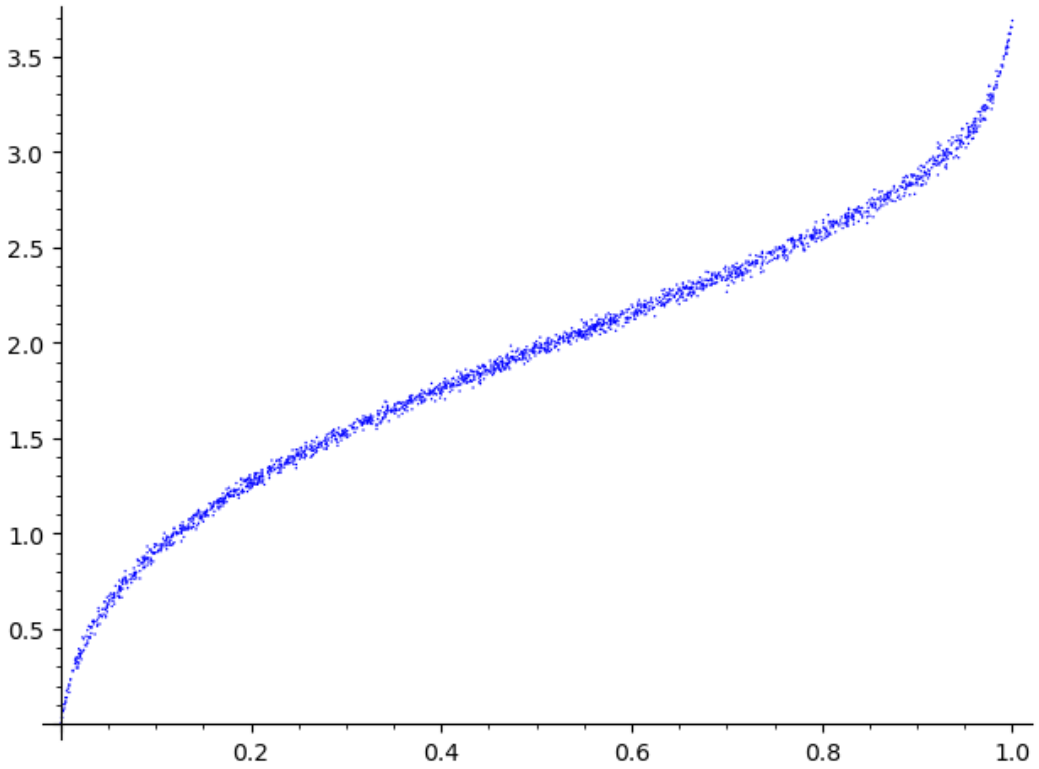
To get an idea how graph entropy typically changes in-between, let us start from the empty graph on N vertices and add edges randomly one by one, computing graph entropy (for uniform X) after each step. The results are plotted below for $N = 10, 20, 40$.







In a similar experiment, we plotted graph entropy (again for uniform X) and edge density for 2000 random graphs[1] on $N = 40$ vertices:



1.3. **The independent set polytope and a simple formula.** In Körner's original paper [5] a fairly simple (non-asymptotic) formula was given for $H_G(X)$. Later, Csiszár, Körner, Lovász, Marton, and Simonyi [1] found an even simpler one, which is based on the following (high-dimensional) polytope.

Definition. Given a simple finite graph G, we say that a subset J of the vertex set $V(G)$ is an *independent set* of G if the induced subgraph $G[J]$ contains no edges. By $\mathcal{J}(G)$ we denote the set of independent sets of G.

The *independent set polytope* or *vertex-packing polytope* $\mathrm{VP}(G)$ is defined as the convex hull of the characteristic vectors of the independent sets of G:

$$\mathrm{VP}(G) \coloneqq \mathrm{conv}\left(\{\mathbb{1}_J \,:\, J \in \mathcal{J}(G)\}\right) \subseteq [0,1]^{V(G)} \subset \mathbb{R}^{V(G)}.$$

Next we consider the following function:

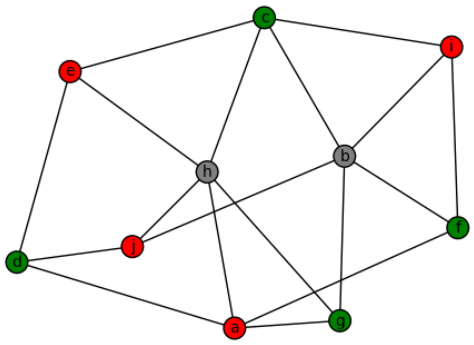
$$(a_x) \mapsto \sum_{x \in V(G)} -p_x \log(a_x) \quad \text{for } \left(a_x\right)_{x \in \mathcal{X}} \in [0,1]^{\mathcal{X}}. \tag{1}$$

It turns out [1] that graph entropy is the minimum of this function over $\mathrm{VP}(G)$:

$$H_G(X) = \min_{(a_x) \in \mathrm{VP}(G)} \sum_{x \in V(G)} -p_x \log(a_x). \tag{2}$$

We explain the details through the following example. Consider the graph below.

[1] We considered Erdős–Rényi random graphs $G(N,p)$ for randomly chosen $p \in [0,1]$.



The vertex set is $V(G) = \{a, b, c, d, e, f, g, h, i, j\}$. Note that the red vertices form an independent set $\{a, e, i, j\}$. For simplicity, we will refer to this set as $aeij$. Similarly, $cdfg$ is also an independent set. In fact, this graph has 11 inclusion-maximal[2] independent sets:

$$acj,\ aeij,\ abe,\ cfgj,\ egij,\ efgj,\ dgi,\ dhi,\ bdh,\ cdfg,\ dfh.$$

The characteristic vector $\mathbb{1}_J$ of an independent set J has ones at the coordinates corresponding to the vertices in J, and has zeros elsewhere. For instance, the vectors corresponding to the sets $aeij$, $cdfg$, bdh, $cfgj$ are the following:

$$\begin{aligned} aeij &: \overset{a}{1}, \overset{b}{0}, \overset{c}{0}, \overset{d}{0}, \overset{e}{1}, \overset{f}{0}, \overset{g}{0}, \overset{h}{0}, \overset{i}{1}, \overset{j}{1}) \\ cdfg &: (0, 0, 1, 1, 0, 1, 1, 0, 0, 0) \\ bdh &: (0, 1, 0, 1, 0, 0, 0, 1, 0, 0) \\ cfgj &: (0, 0, 1, 0, 0, 1, 1, 0, 0, 1) \end{aligned}$$

The vertex-packing polytope $\mathrm{VP}(G)$ consists of points that are convex combinations of such characteristic vectors. For example, if we take the linear combination of the four vectors above with the nonnegative weights $0.4, 0.3, 0.2, 0.1$ (note that their sum is 1), then we get the following point:

$$\left(\overset{a}{0.4}, \overset{b}{0.2}, \overset{c}{0.4}, \overset{d}{0.5}, \overset{e}{0.4}, \overset{f}{0.4}, \overset{g}{0.4}, \overset{h}{0.2}, \overset{i}{0.4}, \overset{j}{0.5}\right) \in \mathrm{VP}(G). \tag{3}$$

For a uniform random X (i.e., when $p_x = 1/10$ for each x), then the function value (1) at this point is:

$$\frac{1}{10}\big(6\log(5/2) + 2\log(5) + 2\log(2)\big) \approx 1.01029.$$

This is, of course, only one possible convex combination. We will later see that there is a simple iterative algorithm for finding the optimal weights. For this particular graph, the optimal weights are $\frac{3}{8}, \frac{1}{4}, \frac{1}{4}, \frac{1}{8}$ (and 0 for the seven remaining maximal independent sets). The corresponding (optimal) point in $\mathrm{VP}(G)$ is

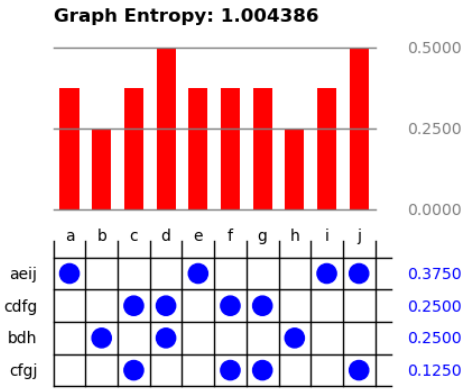
$$\left(\overset{a}{3/8}, \overset{b}{1/4}, \overset{c}{3/8}, \overset{d}{1/2}, \overset{e}{3/8}, \overset{f}{3/8}, \overset{g}{3/8}, \overset{h}{1/4}, \overset{i}{3/8}, \overset{j}{1/2}\right) \in \mathrm{VP}(G). \tag{4}$$

The value at this point gives graph entropy:

$$H_G(X) = \frac{1}{10}\big(6\log(8/3) + 2\log(4) + 2\log(2)\big) \approx 1.00438586.$$

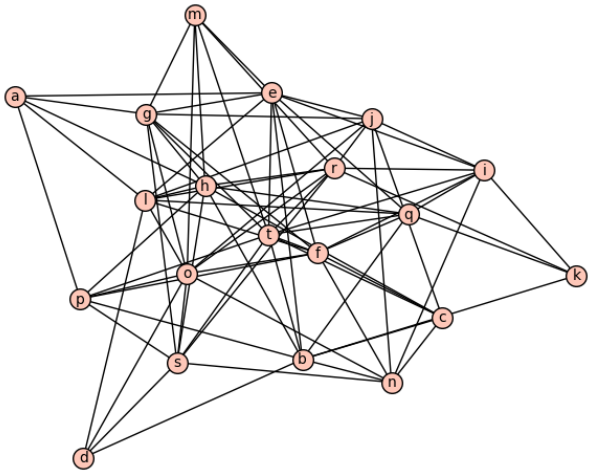
[2]When we want to find the optimal convex combination, then it suffices to consider the inclusion-maximal independent sets (not contained by larger independent sets).

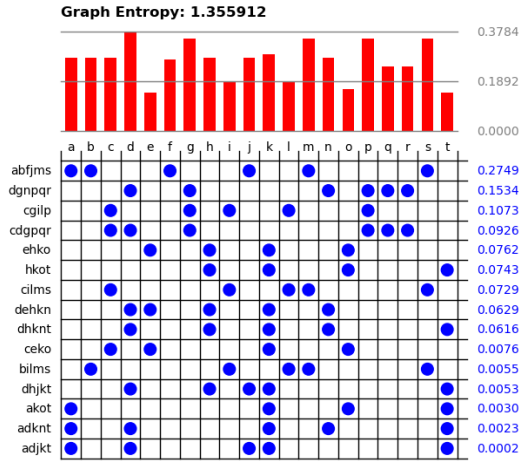
This is summarized by the following graphical representation (that our program code generates automatically when graph entropy is computed):



The rows represent independent sets with positive weights, while the columns correspond to the vertices of G (i.e., the letters of our alphabet). The red bar chart shows the corresponding point (a_x). Given a letter $x \in \mathcal{X} = V(G)$, the height of the bar above x is equal to the coordinate a_x, which is simply the sum of the weights of the sets containing x (that is, the weights at the end of those rows which contain a blue dot $\bullet$ at the column of x).

For a larger example, the graph below has 63 maximal independent sets, out of which 15 are used with positive weights in the optimal convex combination.





1.4. **Optimality check and error bound.** There is a very simple way to check whether a given point $(a_x) \in \mathrm{VP}(G)$ is optimal.

Theorem. *Given* $(a_x) \in \mathrm{VP}(G)$, *if*

$$\sum_{x \in J} \frac{p_x}{a_x} \leq 1 \text{ for all } J \in \mathcal{J}(G),$$

then (a_x) *is optimal:* $H_G(X) = \sum_{x \in V(G)} -p_x \log(a_x)$.

As an illustration, let us confirm that the point given by (4) is indeed optimal: for each independent set J, we need to add up p_x/a_x for $x \in J$, and verify that each sum is at most

one:

$$\begin{aligned}
acj&: \quad (8/3+8/3+2)/10 = \mathbf{11/15};\\
aeij&: \quad (8/3+8/3+8/3+2)/10 = \mathbf{1};\\
abe&: \quad (8/3+4+8/3)/10 = \mathbf{14/15};\\
cfgj&: \quad (8/3+8/3+8/3+2)/10 = \mathbf{1};\\
egij&: \quad (8/3+8/3+8/3+2)/10 = \mathbf{1};\\
efgj&: \quad (8/3+8/3+8/3+2)/10 = \mathbf{1};\\
dgi&: \quad (2+8/3+8/3)/10 = \mathbf{11/15};\\
dhi&: \quad (2+4+8/3)/10 = \mathbf{13/15};\\
bdh&: \quad (4+2+4)/10 = \mathbf{1};\\
cdfg&: \quad (8/3+2+8/3+8/3)/10 = \mathbf{1};\\
dfh&: \quad (2+8/3+4)/10 = \mathbf{13/15}.
\end{aligned}$$

In fact, this optimality check is a special case of the following error bound. Given an arbitrary point $(a_x) \in \mathrm{VP}(G)$, let us compute the following quantity:

$$\delta := \max_{J \in \mathcal{J}} \left(\sum_{x \in J} \frac{p_x}{a_x} - 1 \right).$$

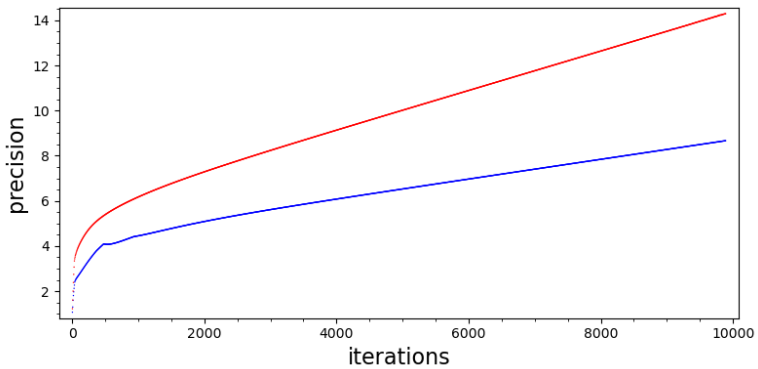
Then the error[3] at (a_x) is at most δ:

$$\left(\sum_{x \in V(G)} -p_x \log(a_x) \right) - H_G(X) \in \big[0, \delta\big].$$

In particular, δ must always be nonnegative, and if $\delta = 0$, then (a_x) must be optimal (leading to the optimality check described in the theorem above).

As an example, consider the point (a_x) as in (3). It is not optimal because for $J = bdh$ we have $(5+2+5)/10 = 1.2$, which is greater than 1. In fact, for this point $\delta = 1.2 - 1 = 0.2$, guaranteeing that $H_G(X)$ is at most $\delta = 0.2$ away from the value (1.01029) at (a_x). The true error for this point is roughly $1.01029 - 1.00439 = 0.0059$.

Later we will see that there is a simple iterative algorithm that always converges to graph entropy. However, the rate of convergence may vary. Computing the error bound above, we are guaranteed to be at most δ away from the actual value of $H_G(X)$. The next plot compares the error bound (blue) to the true error (red) through iterations when we run the algorithm for the cycle graph $G = C_{11}$ with some non-uniform X. (Precision means the number of "precise decimal digits", that is, $-\log_{10}$ of the error.)



[3]By the error at (a_x) we mean the distance of the value at (a_x) from the graph entropy $H_G(X)$.

2. Graph entropy: as an alternating optimization problem

2.1. **Körner's original formula.** An independent set of G contains no edges, and hence any two letters in the set are indistinguishable. Therefore one possible strategy for encoding a sequence $X_1, \dots, X_\ell$ is to replace each X_i with an independent set $J_i \subseteq \mathcal{X}$ containing X_i, and encode the sequence $J_1, \dots, J_\ell$ instead. If we do this randomly in a way that (X_i, J_i) are IID samples of some (X, J) where J is a random independent set containing X, then we can encode the J_i sequence with rate $H(J)$. Note that the number of times any given typical X_i sequence is "covered" by typical J_i sequences has exponential rate $H(J|X)$. Based on this, one can design an encoding with rate $H(J) - H(J|X) = I(X;J)$. Then, for a given X, one needs to choose (X, J) in a way that the mutual information $I(X;J)$ is as small as possible. Körner showed [5] that this is the best achievable code rate, and hence we have the following formula for graph entropy:

$$H_G(X) = \min_{X \in J \text{ ind.set}} I(X;J). \tag{5}$$

2.2. **Alternating optimization hidden in the background.** We have seen that graph entropy can be obtained as the solution of two different minimization problems; see (2) and (5). In fact, they both stem from the same alternating optimization problem that we introduce next. (This was pointed out in [3] in the more general setting of conditional graph entropy.)

Problem. Suppose that we have probability measures on a given finite set $\mathcal{J}$:

- a finite family: μ_x, $x \in \mathcal{X}$;
- and a single measure: ν.

For each $x \in \mathcal{X}$ we have the constraint that the support $\operatorname{supp} \mu_x$ must be contained in a given subset $\mathcal{J}_x$ of $\mathcal{J}$. Find the measures μ_x, ν that minimize the weighted sum of the Kullback–Leibler divergences:

$$\sum_x p_x D_{\mathrm{KL}}(\mu_x \parallel \nu) \text{ for some given weights } p_x \geq 0.$$

To summarize, given $\mathcal{J}_x \subseteq \mathcal{J}$, $x \in \mathcal{X}$ and $p_x \geq 0$, $x \in \mathcal{X}$, find the minimum of the above sum under the constraint $\operatorname{supp} \mu_x \subseteq \mathcal{J}_x$.

In our setting we have a random variable X taking values in the finite set $\mathcal{X}$, and G is a graph on the vertex set $\mathcal{X}$. From this point on we will use small j to denote an independent set of G, hence each j is a subset of $\mathcal{X}$. We choose $\mathcal{J}$ to be the set of all j, while

$$\mathcal{J}_x \coloneqq \{j \,:\, x \in j\}$$

consists of the independent sets containing a fixed x. With this setup and with $p_x \coloneqq \mathbb{P}(X = x)$, the minimum of the problem above turns out to be precisely $H_G(X)$.

To get concrete formulas, let us represent the distributions μ_x and ν by the following vectors:

$$\boldsymbol{q} = \big(q_{j|x}\big)_{(j,x) \in \mathcal{J} \times \mathcal{X}} \in \mathbb{R}^{\mathcal{J} \times \mathcal{X}};$$
$$\boldsymbol{r} = \big(r_j\big)_{j \in \mathcal{J}} \in \mathbb{R}^{\mathcal{J}},$$

where $q_{j|x}$ and r_j stand for $\mu_x(\{j\})$ and $\nu(\{j\})$, respectively.[4]

[4]We index the coordinates/variables by $j|x$ to emphasize the fact that they express certain conditional probabilities. This notation may also serve as a reminder that $q_{j|x}$ have to sum up to 1 for any fixed x.

Then

$$D_{\mathrm{KL}}(\mu_x \parallel \nu) = \sum_j q_{j|x} \log \frac{q_{j|x}}{r_j},$$

and hence the function to minimize is

$$\varphi(\boldsymbol{q}, \boldsymbol{r}) \coloneqq \sum_{x,j} p_x \, q_{j|x} \log \frac{q_{j|x}}{r_j}.$$

The constraints for $\boldsymbol{q}$ and $\boldsymbol{r}$ lead to the following definition. Let $K_{\mathrm{q}} \subset \mathbb{R}^{\mathcal{J}\times\mathcal{X}}$ and $K_{\mathrm{r}} \subset \mathbb{R}^{\mathcal{J}}$ be the following convex polytopes:

$$K_{\mathrm{q}} \coloneqq \left\{ \boldsymbol{q} = \big(q_{j|x}\big) \,:\, q_{j|x} \geq 0; \sum_{j \ni x} q_{j|x} = 1 \,(\forall x \in \mathcal{X});\, q_{j|x} = 0 \text{ if } x \notin j \right\}$$

and

$$K_{\mathrm{r}} \coloneqq \left\{ \boldsymbol{r} = \big(r_j\big) \,:\, r_j \geq 0; \sum_j r_j = 1 \right\}.$$

By $\mathrm{int}(K_{\mathrm{q}})$ and $\mathrm{int}(K_{\mathrm{r}})$ we denote the relative interiors of the polytopes (within their affine hull).

With these notation, we need to find $\min_{K_{\mathrm{q}}\times K_{\mathrm{r}}} \varphi(\boldsymbol{q}, \boldsymbol{r})$. This is an alternating minimization problem: the point is that if we fix one of the two variables $\boldsymbol{q}$ and $\boldsymbol{r}$, then there are explicit formulas for the optimal choice of the other variable.

We define the mappings[5]

$$\begin{aligned} A &\colon K_{\mathrm{r}} \to \mathbb{R}^{\mathcal{X}}; \\ Q &\colon K_{\mathrm{r}} \to K_{\mathrm{q}} \subset \mathbb{R}^{\mathcal{J}\times\mathcal{X}}; \\ R &\colon K_{\mathrm{q}} \to K_{\mathrm{r}} \subset \mathbb{R}^{\mathcal{J}} \end{aligned}$$

by the following coordinate-wise functions $Q_{j|x}$, R_j, A_x:

$$R_j(\boldsymbol{q}) \coloneqq \sum_{x \in j} p_x \, q_{j|x};$$

$$A_x(\boldsymbol{r}) \coloneqq \sum_{j \ni x} r_j;$$

$$Q_{j|x}(\boldsymbol{r}) \coloneqq \begin{cases} 0 & \text{if } x \notin j; \\ \frac{r_j}{A_x(\boldsymbol{r})} & \text{if } x \in j. \end{cases}$$

When minimizing $\varphi(\boldsymbol{q}, \boldsymbol{r})$, it turns out that $\boldsymbol{r} = R(\boldsymbol{q})$ is the optimal choice for a fixed $\boldsymbol{q}$, and similarly $\boldsymbol{q} = Q(\boldsymbol{r})$ is optimal for a fixed $\boldsymbol{r}$; that is, for any $\boldsymbol{q}$ and $\boldsymbol{r}$ we have

$$\varphi(\boldsymbol{q}, \boldsymbol{r}) \geq \varphi\big(\boldsymbol{q}, R(\boldsymbol{q})\big) \text{ and } \varphi(\boldsymbol{q}, \boldsymbol{r}) \geq \varphi\big(Q(\boldsymbol{r}), \boldsymbol{r}\big).$$

So we can explicitly define the following functions:

$$\varphi_{\mathrm{q}}(\boldsymbol{q}) \coloneqq \min_{\boldsymbol{r}} \varphi(\boldsymbol{q}, \boldsymbol{r}) = \varphi\big(\boldsymbol{q}, R(\boldsymbol{q})\big) = \sum_x p_x \sum_{j \ni x} q_{j|x} \log q_{j|x} - \sum_j R_j(\boldsymbol{q}) \log R_j(\boldsymbol{q});$$

$$\varphi_{\mathrm{r}}(\boldsymbol{r}) \coloneqq \min_{\boldsymbol{q}} \varphi(\boldsymbol{q}, \boldsymbol{r}) = \varphi\big(Q(\boldsymbol{r}), \boldsymbol{r}\big) = -\sum_x p_x \log A_x(\boldsymbol{r}) = -\sum_x p_x \log \sum_{j \ni x} r_j.$$

[5]It is straightforward to check that $Q(\boldsymbol{r}) \in K_{\mathrm{q}}$ and $R(\boldsymbol{q}) \in K_{\mathrm{r}}$ always hold. Note that Since the formula for $Q_{j|x}$ involves a division by A_x, it only defines Q over the subset $K_{\mathrm{r}}^* \coloneqq K_{\mathrm{r}} \setminus \bigcup_x A_x^{-1}(0)$. For $\boldsymbol{r} \in K_{\mathrm{r}} \setminus K_{\mathrm{r}}^*$ let $Q(\boldsymbol{r})$ be an arbitrary point in $\mathrm{int}(K_{\mathrm{q}})$.

Then, by definition, $\min\limits_{K_\mathrm{q}\times K_\mathrm{r}} \varphi = \min\limits_{K_\mathrm{q}} \varphi_\mathrm{q} = \min\limits_{K_\mathrm{r}} \varphi_\mathrm{r}$. Note that for

$$\varphi_\mathrm{a}(\boldsymbol{a}) \coloneqq -\sum_x p_x \log a_x$$

we clearly have $\varphi_\mathrm{r}(\boldsymbol{r}) = \varphi_\mathrm{a}\big(A(\boldsymbol{r})\big)$. It follows that $\min_{K_\mathrm{r}} \varphi_\mathrm{r} = \min_{K_\mathrm{a}} \varphi_\mathrm{a}$ if we define K_a as the set of $A(\boldsymbol{r})$:

$$K_\mathrm{a} \coloneqq \big\{A(\boldsymbol{r}) \,:\, \boldsymbol{r} \in K_\mathrm{r}\big\}.$$

Note that $A(\boldsymbol{r})$ is the convex combination of the characteristic functions of the sets $j \subseteq \mathcal{X}$ (with weights r_j). So if $\mathcal{J}$ consists of the independent sets of a graph G on $\mathcal{X}$, then $K_\mathrm{a} = \mathrm{VP}(G)$.

So the q-problem $\min \varphi_\mathrm{q}$ is actually equivalent to Körner's original formula (5) while the a-problem (which is just a simple reformulation of the r-problem) gives back (2).

Theorem. *We have the following formulas for graph entropy:*

$$H_G(X) = \min_{K_q\times K_r} \varphi = \min_{K_q} \varphi_q = \min_{K_r} \varphi_r = \min_{K_a} \varphi_a.$$

2.3. **Iterative algorithm.** When trying to find the minimum of $\varphi(\boldsymbol{q},\boldsymbol{r})$, the fact that we can easily optimize in either variable (while the other is fixed) gives rise to the following simple iterative algorithm. Let us start from a point $\boldsymbol{q}^{(0)}$ and apply R and Q alternately:

$$\boldsymbol{q}^{(0)} \overset{R}{\longmapsto} \boldsymbol{r}^{(0)} \overset{Q}{\longmapsto} \boldsymbol{q}^{(1)} \overset{R}{\longmapsto} \boldsymbol{r}^{(1)} \overset{Q}{\longmapsto} \boldsymbol{q}^{(2)} \overset{R}{\longmapsto} \boldsymbol{r}^{(2)} \cdots . \tag{6}$$

The corresponding φ-value decreases at each step:

$$\begin{array}{ccc}
\varphi\big(\boldsymbol{q}^{(0)},\boldsymbol{r}^{(0)}\big) & = & \varphi_\mathrm{q}(\boldsymbol{q}^{(0)}) \\
\mathbb{V} & & \mathbb{V} \\
\varphi\big(\boldsymbol{q}^{(1)},\boldsymbol{r}^{(0)}\big) & = & \varphi_\mathrm{r}(\boldsymbol{r}^{(0)}) \\
\mathbb{V} & & \mathbb{V} \\
\varphi\big(\boldsymbol{q}^{(1)},\boldsymbol{r}^{(1)}\big) & = & \varphi_\mathrm{q}(\boldsymbol{q}^{(1)}) \\
\mathbb{V} & & \mathbb{V} \\
\varphi\big(\boldsymbol{q}^{(2)},\boldsymbol{r}^{(1)}\big) & = & \varphi_\mathrm{r}(\boldsymbol{r}^{(1)}) \\
\mathbb{V} & & \mathbb{V} \\
\varphi\big(\boldsymbol{q}^{(2)},\boldsymbol{r}^{(2)}\big) & = & \varphi_\mathrm{q}(\boldsymbol{q}^{(2)}) \\
\vdots & & \vdots
\end{array}$$

One can also think of this alternating optimization as "jumping" between the q-problem $\min_{K_\mathrm{q}} \varphi_\mathrm{q}$ and the r-problem $\min_{K_\mathrm{r}} \varphi_\mathrm{r}$ using the maps $Q\colon K_\mathrm{r} \to K_\mathrm{q}$ and $R\colon K_\mathrm{q} \to K_\mathrm{r}$. The value to minimize (i.e., the φ_q-value and the φ_r-value, respectively) always decreases, so with each step we get closer to the optimum.

Following the footsteps of the general theory of Csiszár and Tusnády [2], it was shown in [3] that for an arbitrary starting point $\boldsymbol{q}^{(0)}$ in the relative interior $\mathrm{int}(K_\mathrm{q})$, the iterative process converges to the minimum.

Theorem ([3])**.** *For an arbitrary starting point $\boldsymbol{q}^{(0)} \in \mathrm{int}(K_q)$ consider the sequence* (6) *obtained by alternating optimization. Then $\varphi\big(\boldsymbol{q}^{(n)},\boldsymbol{r}^{(n)}\big)$ is a decreasing sequence that converges to $\min\limits_{K_q\times K_r} \varphi = H_G(X)$ as $n \to \infty$.*

3. Conditional graph entropy

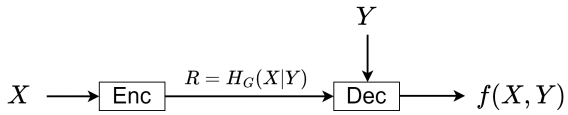
3.1. **The generalization of Körner's formula.** The analogous problem with side information Y_i at the receiver leads to the notion of *conditional graph entropy* $H_G(X|Y)$. Let (X,Y) be discrete random variables of some given joint distribution and let (X_i,Y_i) be IID

samples. We assume that the decoder knows the sequence $Y_1, Y_2, \ldots$, and, as before, that some of the outcomes of X are indistinguishable, described by a graph G. If we want to use the same approach (i.e., choosing a random independent set J), then J and Y should be independent conditioned on X (because the sender does not know Y_i when choosing J_i). This can be made rigorous, leading to the following formula:

$$H_G(X|Y) = \min_J I(X; J \,|\, Y) = \min_J \Bigg(H(J|Y) - \underbrace{H(J|X,Y)}_{H(J|X)} \Bigg),$$

where J is a random independent set of G such that $X \in J$, and J and Y are conditionally independent conditioned on X (which is equivalent to saying that $Y - X - J$ is a Markov chain).

3.2. **Compression with side information.** Suppose now that the receiver wishes to recover the values $f(X_i, Y_i)$ of some given function $f\colon \mathcal{X} \times \mathcal{Y} \to \mathcal{Z}$ (with high probability, over long blocks) as depicted in the figure below.



Orlitsky and Roche [6] showed that the minimal rate of information that needs to be transmitted is precisely the conditional graph entropy of the so-called *characteristic graph*, which is defined on the vertex set $\mathcal{X}$ as follows: vertices $x_1, x_2 \in \mathcal{X}$ are connected with an edge if and only if

$$\exists y \in \mathcal{Y} \text{ s.t. } \big(f(x_1, y) \neq f(x_2, y) \;\&\; \mathbb{P}(X = x_1, Y = y) > 0 \;\&\; \mathbb{P}(X = x_2, Y = y) > 0\big).$$

We mention that in the special case $f(x, y) = x$, which was already studied in Shannon's classical work [7], the optimal rate is given by the conditional entropy $H(X|Y) = H(X, Y) - H(Y)$.

3.3. **Generalized formulas.** In this conditional setting the alternating optimization problem is as follows.

Problem. Suppose that we have two finite families of probability measures on a given finite set $\mathcal{J}$: μ_x, $x \in \mathcal{X}$ and ν_y, $y \in \mathcal{Y}$. In the first family for each $x \in \mathcal{X}$ we have a constraint: the support $\operatorname{supp} \mu_x$ must be contained in a given subset $\mathcal{J}_x$ of $\mathcal{J}$. Find the measures μ_x, ν_y that minimize the weighted sum of the Kullback–Leibler divergences:

$$\sum_{x,y} p_{x,y} D_{\mathrm{KL}}(\mu_x \parallel \nu_y) \text{ for some given weights } p_{x,y} \geq 0.$$

To summarize, given $\mathcal{J}_x \subseteq \mathcal{J}$, $x \in \mathcal{X}$ and $p_{x,y} \geq 0$, $x \in \mathcal{X}, y \in \mathcal{Y}$, find the minimum of the above sum under the constraint $\operatorname{supp} \mu_x \subseteq \mathcal{J}_x$.

In our setting we have random variables X and Y taking values in the finite sets $\mathcal{X}$ and $\mathcal{Y}$, respectively, and G is a graph on the vertex set $\mathcal{X}$. As before, $\mathcal{J}$ is the set of all independent sets j of G, while

$$\mathcal{J}_x \coloneqq \{j \,:\, x \in j\}$$

consists of the independent sets containing a fixed x. With $p_{x,y} \coloneqq \mathbb{P}(X = x, Y = y)$, the above minimum is $H_G(X|Y)$.

Let us represent the distributions μ_x and ν_y by the following vectors:

$$\boldsymbol{q} = \big(q_{j|x}\big)_{(j,x)\in\mathcal{J}\times\mathcal{X}} \in \mathbb{R}^{\mathcal{J}\times\mathcal{X}};$$
$$\boldsymbol{r} = \big(r_{j|y}\big)_{(j,y)\in\mathcal{J}\times\mathcal{Y}} \in \mathbb{R}^{\mathcal{J}\times\mathcal{Y}},$$

where $q_{j|x}$ and $r_{j|y}$ stand for $\mu_x(\{j\})$ and $\nu_y(\{j\})$, respectively. Then

$$D_{\mathrm{KL}}(\mu_x \parallel \nu_y) = \sum_j q_{j|x} \log \frac{q_{j|x}}{r_{j|y}},$$

so the function to minimize is

$$\varphi(\boldsymbol{q}, \boldsymbol{r}) \coloneqq \sum_{x,y,j} p_{x,y}\, q_{j|x} \log \frac{q_{j|x}}{r_{j|y}}.$$

The constraints for $\boldsymbol{q}$ and $\boldsymbol{r}$ lead to the following definitions:

$$K_{\mathrm{q}} \coloneqq \left\{ \boldsymbol{q} = \big(q_{j|x}\big) \,:\, q_{j|x} \ge 0;\ \sum_{j\ni x} q_{j|x} = 1\ (\forall x \in \mathcal{X});\ q_{j|x} = 0 \text{ if } x \notin j \right\}$$

and

$$K_{\mathrm{r}} \coloneqq \left\{ \boldsymbol{r} = \big(r_{j|y}\big) \,:\, r_{j|y} \ge 0;\ \sum_{j} r_{j|y} = 1\ (\forall y \in \mathcal{Y}) \right\}.$$

The general formulas for the mappings $A\colon K_{\mathrm{r}} \to \mathbb{R}^{\mathcal{X}}$; $Q\colon K_{\mathrm{r}} \to K_{\mathrm{q}} \subset \mathbb{R}^{\mathcal{J}\times\mathcal{X}}$; $R\colon K_{\mathrm{q}} \to K_{\mathrm{r}} \subset \mathbb{R}^{\mathcal{J}\times\mathcal{Y}}$ are as follows:

$$R_{j|y}(\boldsymbol{q}) \coloneqq \sum_{x\in j} p_{x|y}\, q_{j|x};$$

$$A_x(\boldsymbol{r}) \coloneqq \sum_{j\ni x} \prod_y \big(r_{j|y}\big)^{p^{y|x}};$$

$$Q_{j|x}(\boldsymbol{r}) \coloneqq \begin{cases} 0 & \text{if } x \notin j; \\ \prod_y \big(r_{j|y}\big)^{p^{y|x}} \Big/ A_x(\boldsymbol{r}) & \text{if } x \in j. \end{cases}$$

(Here we define $t^0 = 1$ even for $t = 0$.)

Remark. Note that R is a linear map and it actually describes how the conditional distributions $J \,|\, Y = y$ can be expressed in terms of $J \,|\, X = x$ in a Markov chain $Y - X - J$.

As for φ_{q} and φ_{r}, we have

$$\varphi_{\mathrm{q}}(\boldsymbol{q}) \coloneqq \varphi\big(\boldsymbol{q}, R(\boldsymbol{q})\big) = \sum_x p_x \sum_{j\ni x} q_{j|x} \log q_{j|x} - \sum_y p^y \sum_j R_{j|y}(\boldsymbol{q}) \log R_{j|y}(\boldsymbol{q})$$

and

$$\varphi_{\mathrm{r}}(\boldsymbol{r}) \coloneqq \varphi\big(Q(\boldsymbol{r}), \boldsymbol{r}\big) = -\sum_x p_x \log A_x(\boldsymbol{r}) = -\sum_x p_x \log \sum_{j\ni x} \prod_y \big(r_{j|y}\big)^{p^{y|x}}.$$

With these notations, we can express conditional graph entropy in various ways:

$$H_G(X|Y) = \min_{K_{\mathrm{q}}} \varphi_{\mathrm{q}} = \min_{K_{\mathrm{r}}} \varphi_{\mathrm{r}} = \min_{K_{\mathrm{r}}} -\sum_x p_x \log A_x.$$

It can be seen easily that $A_x\colon K_\mathrm{r} \to [0,1]$ is a concave function for each x. It means that the set of image points $\boldsymbol{a} = (a_x)_{x\in\mathcal{X}}$ with $a_x = A_x(\boldsymbol{r})$, as $\boldsymbol{r}$ ranges over K_r, (essentially) defines a convex corner[6] K_a in $\mathbb{R}^{\mathcal{X}}$. Then we have

$$H_G(X|Y) = \min_{K_\mathrm{a}} \varphi_\mathrm{a}, \text{ where } \varphi_\mathrm{a}(\boldsymbol{a}) = -\sum_x p_x \log a_x.$$

A nice feature of this a-problem is that the minimum is attained at a single point $\boldsymbol{a} \in K_\mathrm{a}$ because φ_a is strictly convex (provided that $p_x = \mathbb{P}(X = x) > 0$ for each x). Also note that φ_a depends only on the distribution of X, while the convex corner K_a depends only on the graph G and the conditional distributions $Y \,|\, X = x$ for any given x. Thus, the parameters of the problem are, so to say, split between φ_a and K_a.

Moreover, in [3] another convex corner, denoted by L, was defined such that

$$H_G(X|Y) = \min_{K_\mathrm{a}} \varphi_\mathrm{a} = -\min_{L} \varphi_\mathrm{a};$$

a vector $\boldsymbol{a} = \big(a_x\big)_{x\in\mathcal{X}} \in K_\mathrm{a}$ being optimal (i.e., the minimum point of φ_a) if and only if $\boldsymbol{a}^{-1} := \big(a_x^{-1}\big)_{x\in\mathcal{X}} \in L$. This provides a fairly simple way to check optimality, and even leading to an error bound for the iterative algorithm, which was shown to converge to $H_G(X|Y)$ in this conditional setting as well.

Further material

For a more detailed exposition of the conditional setting, see [3]. Simonyi's excellent survey [8] is a good source for further results on graph entropy. A possible generalization to graphons can be found in [4].

References

[1] Imre Csiszár, János Körner, László Lovász, Katalin Marton, and Gábor Simonyi. Entropy splitting for antiblocking corners and perfect graphs. *Combinatorica*, 10(1):27–40, 1990.

[2] Imre Csiszár and Gábor Tusnády. Information geometry and alternating minimization procedures. *Statistics and Decisions*, Supp. 1:205–237, 1984.

[3] Viktor Harangi, Xueyan Niu, and Bo Bai. Conditional graph entropy as an alternating minimization problem, 2022.

[4] Viktor Harangi, Xueyan Niu, and Bo Bai. Generalizing körner's graph entropy to graphons, 2022.

[5] János Körner. Coding of an information source having ambiguous alphabet and the entropy of graphs. In *6th Prague conference on information theory*, pages 411–425, 1973.

[6] Alon Orlitsky and James R. Roche. Coding for computing. *IEEE Transactions on Information Theory*, 47:903–917, 1998.

[7] Claude E. Shannon and Warren Weaver. *The Mathematical Theory of Communication*. University of Illinois Press, Urbana and Chicago, 1949.

[8] Gábor Simonyi. Perfect graphs and graph entropy. An updated survey. In Jorge Ramirez-Alfonsin and Bruce Reed, editors, *Perfect Graphs*, pages 293–328. John Wiley and Sons, 2001.

[6] A convex corner of $\mathbb{R}^{\mathcal{X}}$ is a convex compact set in the positive orthant $[0,\infty)^{\mathcal{X}}$ that is *downward closed*, i.e., if we take any point in the set and decrease some of its coordinates, then the new point still lies in the set.