1 Information-theoretic Interpretations

by: Isabeau Prémont-Schwarz

This post is written for the AIHelsinki study group on natural image statistics. The content of the post is based on chapter 8 of the Natural Image Statistics book which is available here `http://www.naturalimagestatistics.net/` and the presentation of this chapter by Vikas Verma, a few derivations have been added. The post is intended as a summary of what was learned at the AIHelsinki study group on the topic of Information-theoretic interpretations.

We will first talk about the theory before going to some applications. In the theory part we will cover the basic motivation, entropy as a measure of uncertainty, and mutual information. In the applications part we will cover mutual information as used for sparse coding and infomax with nonlinear neurons.

1.1 Theory

1.1.1 Motivation

Information theory was first developed in the context of signal transmission. Imagine that we have to transmit messages of the form

BABABABADABACAABAACABDAAAAABAAAAAAAADBCA

composed of the characters A, B, C, D. With each of the characters having the following probabilities to appear at any given position:

$p(A) = 1/2$ $p(B) = 1/4$

$p(C) = 1/8$ $p(D) = 1/8$

Now let's assume we need to code these messages in binary (with 0s and 1s). A naive way of doing this would be to use 2 bits for each letter:

$A \to 00$ $B \to 01$ $C \to 10$ $D \to 11$

This would mean that our 40-letter message would be encoded using 80 bits. But actually we encode our messages more efficiently if instead we use fewer bits for more common letters and more bits for rarer letters. For example, if instead we use one bit for A, 2 bits for B and 3 bits for C and D, for instance:

$A \to 0$ $B \to 10$ $C \to 110$ $D \to 111$

Using this encoding, a typical 40-letter message can be encoded using

$$40 \cdot \big(p(A) \times 1 + p(B) * 2 + p(C) * 3 + p(D) * 3\big) = 40(1/2 + 1/2 + 3/8 + 3/8) = 70 bits$$

Thus using this encoding would require 12.5% fewer bits to send the same message. This is actually (as we will see below) the optimal way (using the least amount of bits) of coding our messages in binary. This is what (Shannon) entropy is: *Entropy is the average amount of bits necessary to send a symbol of your message using optimal coding.* Or said another way: *Entropy is the minimum amount of bits necessary (on average) required to send symbols in your message.*

This means that, denoting the number of bits required to encode X by $l(X)$, we have that the entropy of letter sequences is:

$$H = \mathbb{E}(l(X)) = p(A)l(A) + p(B)l(B) + p(C)l(C) + p(D)l(D) = \\ 1/2 * 1 + 1/4 * 2 + 1/8 * 3 + 1/8 * 3 = 1.75. \quad (1)$$

Thus, on average our letters contain 1.75 bits of information and so its entropy is 1.75.

So how do we calculate this Entropy? How do we figure out what is the optimal encoding?

1.1.2 Entropy as Minimum Coding Length

If we define the entropy as the minimum amount of bits required in average for all our symbols, we need to calculate what is the minimum coding length for a given code.

Finding the minimum coding length, is a simple optimization problem. Assume we have an alphabet of N symbols X_i with $i \in [1, N]$ and symbol X_i has a probability $p(X_i) = p_i$ of appearing at any given position in a message. We further denote $l(X_i) = l_i$ to be the coding length in number of bits for symbol X_i.

What is important to notice at this point is that we can't have 3 different symbols coded with one bit each. We only have 0 and 1. So there is we can have max 2 different symbols coded with one bit. And if we do have two symbols coded with a single bit, say $A \to 0$ and $B \to 1$, then we cannot represent any other symbol because A and B already occupy the whole coding space, 0001110 already means AAABBBA. So a symbol coded with 1 bit takes up 1/2 of the coding space. And in general a symbol coded with l bits takes up 2^{-l} of the coding space. So we are constrained by

$$\sum_{i=0}^{N} 2^{-l_i} \leq 1. \quad (2)$$

But to be optimal we wish to use the whole coding space so we want to satisfy the constraint

$$\sum_{i=0}^{N} 2^{-l_i} = 1. \quad (3)$$

So out goal is to minimize

$$\mathbb{E}(l_i) = \sum_{i=0}^{N} p_i l_i, \quad (4)$$

under constraint (3). And so using a Lagrange multiplier λ to enforce the constraint we need to extremize:

$$\sum_{i=0}^{N} p_i l_i + \lambda\big(\sum_{i=0}^{N} 2^{-l_i} - 1\big). \quad (5)$$

Which means we get differentiating with respect to l_i:

$$p_i - \lambda \ln(2) 2^{-l_i} = 0. \tag{6}$$

And so solving for l_i we get

$$l_i = -\log_2(p_i). \tag{7}$$

This means that for optimal minimal length coding, a symbol X should be coded using $\log_2(p(X))$ bits. We can now check that our second coding in the above section was optimal. We indeed had:

$$l(A) = 1 = -\log_2(1/2) = \log_2(p(A)) \tag{8}$$
$$l(B) = 2 = -\log_2(1/4) = \log_2(p(B)) \tag{9}$$
$$l(C) = 3 = -\log_2(1/8) = \log_2(p(C)) \tag{10}$$
$$l(A) = 3 = -\log_2(1/8) = \log_2(p(D)). \tag{11}$$

We can plugin our new-found result (7) in (1) to obtain:

$$H = \sum_{i=0}^{N} p_i l_i = -\sum_{i=0}^{N} p_i \log_2(p_i). \tag{12}$$

And this is how we will actually define the entropy. Notice that in sum case, it might be impossible to realize the minimal coding length in practice because it would require a fractional number of bits for some symbols. But even in those case we will define the entropy to be (12) even if the minimal coding length cannot be achieved in practice. With this definition we have that the entropy is the expected number of bits of information which we will acquire for every new symbol we receive.

1.1.3 Differential Entropy

So far we have talked about discrete random variables, for which we defined the entropy to be

$$H = -\sum_{i=0}^{N} p_i \log_2(p_i), \tag{13}$$

for an entropy calculated in units of bits or

$$H = -\sum_{i=0}^{N} p_i \log_e(p_i) = -\sum_{i=0}^{N} p_i \ln(p_i), \tag{14}$$

for an entropy calculated in units of nats (natural unit of information). Note that since the probabilities p_i are always between 0 and 1, $-\log(p_i)$ is always positive and so entropy is always positive, this will not always be the case in the

continuous case. But how should we define entropy in the continuous random variable case? Well, like one normally does in those cases, we simply replace the sum by an integral:

$$H = -\int_{x=-\infty}^{+\infty} p(x)\log(p(x))dx, \tag{15}$$

where $p(x)$ is now the probability density function.

Back in the discrete case, entropy was somewhat intuitively understood as the number of bits on carried on average by one reading of the value of the random variable and intuition was a good guide as to what to expect. Intuition is not a very good guide anymore in the continuous case. To see this, let us consider the case of the uniform distribution on the interval $[0, \alpha]$:

$$p(x) = \begin{cases} \frac{1}{\alpha}, & \text{if } 0 \leq x \leq \alpha \\ 0, & \text{otherwise.} \end{cases} \tag{16}$$

Which gives us an entropy of:

$$H(x) = -\int_{x=0}^{+\alpha} \frac{1}{\alpha}\log(\frac{1}{\alpha})dx = \log(\alpha). \tag{17}$$

From this we see that depending on the value of α, the entropy can be negative ($\alpha < 1$, zero $\alpha = 1$, or positive $\alpha > 1$, and it ranges from $-\infty$ ($\alpha \to 0^+$) to $+\infty$ ($\alpha \to +\infty$). Conceptually the entropy of a continuous variable can be negative because, compared to the discrete case, we omit an infinite component. We can compare the differential entropy to the discrete entropy by discretizing the continuous variable and calculating the discrete entropy for it. Let us consider a continuous random variable which takes values in the interval [0,1] with probability density function $p(x)$ with $p(x) = 0$ for $x > 1$ or $x < 0$. We transform it into a discrete probability problem by dividing the interval [0,1] into n bins of size $1/n$. Bin i is the interval $[\frac{i-1}{n}, \frac{i}{n}]$. We thus have that the probability p_i that variable falls in bin i is

$$p_i = -\int_{x=\frac{i-1}{n}}^{+\frac{i}{n}} p(x)dx \approx p(\frac{i}{n}) \cdot \frac{1}{n}, \tag{18}$$

where the approximation is valid for a continuous $p(x)$ when the intervals are small enough (n large enough). Therefore the entropy for the discrete process

of ending up in different bins is

$$H_{discrete} = -\sum_{i=0}^{N} p_i \log(p_i) \approx -\sum_{i=0}^{N} p(\frac{i}{n}) \log(\frac{p(\frac{i}{n})}{n}) \cdot \frac{i}{n} \tag{19}$$

$$= -\sum_{i=0}^{N} p(\frac{i}{n})(\log(p(\frac{i}{n})) - \log(n)) \cdot \frac{i}{n} \tag{20}$$

$$\approx -\int_{x=0}^{1} p(x)(\log(p(x)) - \log(n))dx \tag{21}$$

$$= H_{differential} + \log(n). \tag{22}$$

As the continuous case is obtained from the discrete case by taking the limit $n \to \infty$, we see that differential entropy is equal to the discrete entropy minus a constant infinite component (equal to $\log(\aleph_0) = \aleph_0$) which is the same for all continuous distributions. As such, differential entropy should be thought of not as absolute but as relative to other entropy values.

A natural question to ask at this point is what distribution has the maximum entropy. Of the distributions with support on the interval [0,1], the uniform distribution is that with maximal entropy as one would intuitively expect: we gain the most information by reading the value of the random variable if all values were equally likely. But, of the distributions on $[-\infty, +\infty]$ with unit variance, what is the distribution with maximum entropy? That is a less trivial question. We can answer that question using calculus of variations. Further constraining the mean to be 0 (the problem is translation invariant), we wish to maximize

$$H = -\int_{x=-\infty}^{+\infty} p(x)\big(\log(p(x)) + \lambda(x^2 - 1) + \mu + \rho(x - 0)\big)dx + \mu, \tag{23}$$

where the second term with the Lagrange multiplier λ is to impose the unit variance constraint, the Lagrange multiplier μ is to impose the constraint that the integral of $p(x)$ should be one, and the Lagrange multiplier ρ is to impose zero mean. Using calculus of variations we have

$$\delta H = -\int_{x=-\infty}^{+\infty} \delta p(x)\big(\log(p(x)) + \lambda(x^2 - 1) + 1 + \mu + \rho x\big)dx. \tag{24}$$

Thus our distribution must satisfy

$$\log(p(x)) + \lambda(x^2 - 1) + 1 + \mu + \rho x = 0 \tag{25}$$

$$p(x) = \exp\big(-\lambda(x^2 - 1) - 1 - \mu - \rho x\big). \tag{26}$$

Solving for λ, μ, and ρ using the constraints we finally have that

$$p(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}, \tag{27}$$

which is the normal distribution with mean zero and unit variance. We thus conclude that for a given fixed variance, the distribution with the maximum entropy is the Gaussian distribution. Thus in some sense, the Gaussian distribution is the "most random" distribution, the "least deterministic" one. Viewed in the way, the central limit theorem, which says that the average of independent identically distributed variables tends towards the Gaussian in the infinite limit, is not so surprising: adding randomness by adding more independent variables will converge to the most random of all distribution, the Gaussian distribution.

1.1.4 Mutual Information

Suppose you have two random variables x and y and that x and y might not be independent. The probability of x given y, $p(x|y)$, is the conditional probability of x given y. The entropy for this conditional probability distribution is

$$H = -\sum_{i=0}^{N} p(x_i|y_k)\log(p(x_i|y_k)), \tag{28}$$

for some k. If we average this entropy over all possible values of y, we get what is called the *conditional entropy*:

$$H(x|y) = -\sum_{k=0}^{M}\sum_{i=0}^{N} p(y_k)p(x_i|y_k)\log(p(x_i|y_k)), \tag{29}$$

which tells us on average how much more (bits of) information we get on average by knowing the value of x if we already know the value y. Notice that if x and y are independent so that $p(x|y) = p(x)$ then we have that $H(x|y) = H(x)$.

In the above we wrote down the equations for discrete random variables, but exactly the same considerations apply for continuous variable simply by replacing the sum with an integral. In this section, unless explicitly mentioned everything we derive or say will apply equally well to the discrete and continuous case (simply by switching between sums and integrals) unless we say so explicitly.

Knowing that $p(x|y) = \frac{p(x,y)}{p(y)}$ we have that

$$\begin{aligned}
H(x|y) &= -\sum_{k=0}^{M}\sum_{i=0}^{N} p(y_k)p(x_i|y_k)\log(p(x_i|y_k)) && (30)\\
&= -\sum_{k=0}^{M}\sum_{i=0}^{N} p(y_k)\frac{p(x_i,y_k)}{p(y_k)}\log(\frac{p(x_i,y_k)}{p(y_k)}) && (31)\\
&= -\sum_{k=0}^{M}\sum_{i=0}^{N} p(x_i,y_k)\log(p(x_i,y_k)) + \sum_{k=0}^{M}\sum_{i=0}^{N} p(x_i,y_k)\log(p(y_k)) && (32)\\
&= H(x,y) - H(y), && (33)
\end{aligned}$$

where $H(x,y)$ is the entropy of the combined variables x and y.

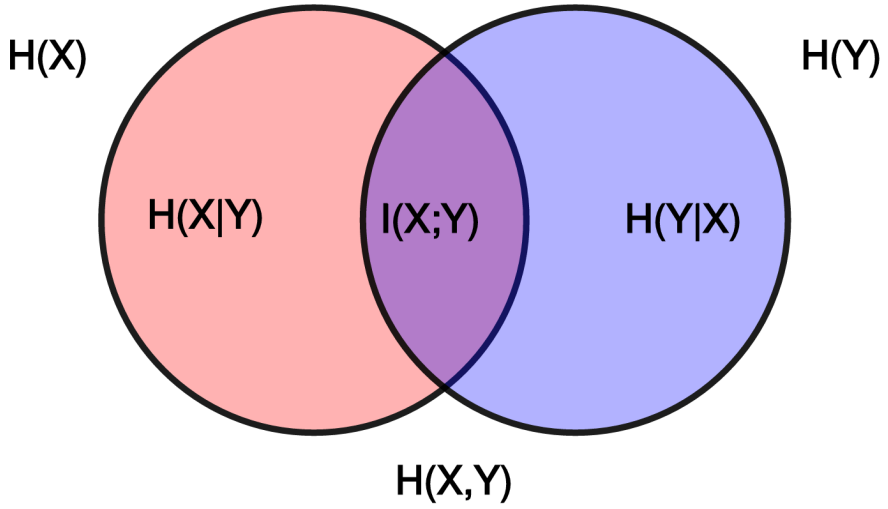

Figure 1: The area contained by both circles is the joint entropy $H(X,Y)$. The circle on the left (red and violet) is the individual entropy $H(X)$, with the red being the conditional entropy $H(X|Y)$. The circle on the right (blue and violet) is $H(Y)$, with the blue being $H(Y|X)$. The violet is the mutual information $I(X;Y)$. Image source: Wikipedia.

Mutual information, $I(x,y)$, between x and y, is the information which is shared by x and y, so it is the information contained in x ($H(x)$) minus the information in x which is not in y ($H(x|y)$):

$$I(x,y) = H(x) - H(x|y) \tag{34}$$
$$= H(x) + H(y) - H(x,y) \tag{35}$$
$$= \Big(-\sum_{k=0}^{M}\sum_{i=0}^{N} p(x_i,y_k)\log(p(x_i)p(y_k))\Big) - \Big(-\sum_{k=0}^{M}\sum_{i=0}^{N} p(x_i,y_k)\log(p(x_i,y_k))\Big) \tag{36}$$
$$= -\sum_{k=0}^{M}\sum_{i=0}^{N} p(x_i,y_k)\log\big(\frac{p(x_i)p(y_k)}{p(x_i,y_k)}\big), \tag{37}$$

where $p(x_i) = \sum_{k=0}^{M} p(x_i,y_k)$ is the marginal distribution of x and $p(y_k) = \sum_{i=0}^{N} p(x_i,y_k)$ is the marginal distribution of y.

At this point it is useful to look at Gibbs' inequality which states that for two probability distributions r and q

$$-\sum_{i=0}^{N} r_i \log(r_i) \leq -\sum_{i=0}^{N} r_i \log(q_i), \tag{38}$$

or in the continuous case that

$$-\int r(x)\log(r(x))dx \leq -\int r(x)\log(q(x)), \tag{39}$$

with equality if and only if $r = q$. By applying Gibbs' inequality to (36), with $r = p(x, y)$ and $q = p(x)p(y)$, we see that the mutual information $I(x, y)$ is always positive and is equal to zero if and only if $p(x, y) = p(x)p(y)$. In other words, this means that the mutual information is non-negative and it is zero if and only if x and y are independent variables. Looking at (34) this also means that

$$H(x) \geq H(x|y), \tag{40}$$

with equality if and only if x and y are independent random variables.

So in conclusion, we see that $H(x, y)$, $H(x)$, $H(x|y)$, and $I(x, y)$ fit together like in the Venn diagram of Fig. 1. But remember that while for discrete variables, all of these components are non-negative, this is not true for the continuous case where $H(x)$, $H(x|y)$, and $H(x, y)$ can all be negative, only $I(x, y)$ is always non-negative even in the continuous case.

1.2 Applications

1.2.1 Sparse Coding

Now that we have developed this these information-theoretic tools, we can look at sparse coding and ICA from an information-theoretic perspective. In sparse coding and ICA we have signals z_i (which are usually whitened) and we want to find sparse sources s_j for these signals by orthogonally rotating the z's:

$$s_j = \sum_i v_{ji} z_i. \tag{41}$$

We denote $H(\boldsymbol{z}) = H(z_1, \ldots, z_N)$ and $\boldsymbol{s} = \boldsymbol{v} \cdot \boldsymbol{z}$. If $\boldsymbol{v}$ is an orthonormal matrix, we are only rotating the probability space so that

$$\begin{aligned} H(s_1, \ldots, s_N) &= H(\boldsymbol{s}) && (42)\\ &= H((\boldsymbol{v} \cdot \boldsymbol{z})_1, \ldots, (\boldsymbol{v} \cdot \boldsymbol{z})_N) && (43)\\ &= H(\boldsymbol{v} \cdot \boldsymbol{z}) && (44)\\ &= H(\boldsymbol{z}) && (45)\\ &= H(z_1, \ldots, z_N). && (46) \end{aligned}$$

So in ICA or in spare coding, if the sparsity function is $h(s^2) = \log(p(s))$, we try to minimize

$$\sum_i H(s_i) = \sum_i H((\boldsymbol{v} \cdot \boldsymbol{z})_i). \tag{47}$$

But in the case in which $\boldsymbol{v}$ is orthonormal, that is the same as minimizing

$$\left(\sum_i H(s_i)\right) - H(\boldsymbol{z}) = \left(\sum_i H(s_i)\right) - H(\boldsymbol{v} \cdot \boldsymbol{z}) \tag{48}$$

$$= \left(\sum_i H(s_i)\right) - H(\boldsymbol{s}). \tag{49}$$

And from the definition of mutual information (equation (34)) we have that

$$\left(\sum_i H(s_i)\right) - H(\boldsymbol{s}) = \sum_{K=2}^{N} \sum_{\{i_1,\ldots,i_K\} \in \{1,\ldots,N\}} (K-1) \cdot I(x_{i_1}, \ldots, x_{i_K} | x_j \text{s.t.} j \notin \{i_1, \ldots, i_K\}), \tag{50}$$

so we see that in this case, both ICA and sparse coding are equivalent to minimizing the mutual information between the sources to have the sources be as independent as possible.

1.2.2 Infomax with Nonlinear Neurons

Let us consider a neuron-network layer which has (nonlinear) activation function ϕ and input vector $\boldsymbol{x}$ which is considered to be continous-valued. Let $\boldsymbol{b}$ be a transformation matrix on $\boldsymbol{x}$. Then we have that the activation of neuron i is

$$y_i = \phi((\boldsymbol{b} \cdot \boldsymbol{x})_i) + n_i, \tag{51}$$

where n is some independent (Gaussian) noise.

The principle behind infomax is to try to keep as much information about $\boldsymbol{x}$ in $\boldsymbol{y}$. Thus we want to maximize the mutual information between $\boldsymbol{x}$ and $\boldsymbol{y}$

$$I(\boldsymbol{x}, \boldsymbol{y}) = H(\boldsymbol{y}) - H(\boldsymbol{y}|\boldsymbol{x}). \tag{52}$$

But because $p(\boldsymbol{y}|\boldsymbol{x}) = p(\boldsymbol{n})$ doesn't depend on $\boldsymbol{b}$ we can forget about the $H(\boldsymbol{y}|\boldsymbol{x})$ and just try to maximize the entropy of $\boldsymbol{y}$, $H(\boldsymbol{y})$.

$$H(\boldsymbol{y}) = -\int p_y(\boldsymbol{y}) \log(p_y(\boldsymbol{y})) d\boldsymbol{y} \tag{53}$$

$$= -\int p_x(\boldsymbol{x}) \log(p_y(\phi(\boldsymbol{b} \cdot \boldsymbol{x}))) d\boldsymbol{x} \tag{54}$$

$$= -\int p_x(\boldsymbol{x}) \left(\log(p_x(\boldsymbol{x})) - (\sum_{i=0}^{N} \log(\phi'((\boldsymbol{b} \cdot \boldsymbol{x})_i))) - \log(\det(\boldsymbol{b})) \right) \tag{55}$$

$$= H(\boldsymbol{x}) + \sum_i \mathbb{E}_x \left(\log(\phi'((\boldsymbol{b} \cdot \boldsymbol{x})_i)) \right) + \log(\det(\boldsymbol{b})). \tag{56}$$

Where we used the fact that

$$p_y(\boldsymbol{y})d\boldsymbol{y} = p_x(\boldsymbol{x})d\boldsymbol{x} \tag{57}$$

$$p_y(\boldsymbol{y}) = p_x(\boldsymbol{x})(\det(\frac{d\boldsymbol{y}}{d\boldsymbol{x}}))^{-1} \tag{58}$$

$$= p_x(\boldsymbol{x}) \cdot \left(\prod_{i=0}^{N} \phi'((\boldsymbol{b} \cdot \boldsymbol{x})_i) \cdot \det(\boldsymbol{b}) \right)^{-1}. \tag{59}$$

Because $H(\boldsymbol{x})$ does not depend on $\boldsymbol{b}$, we can drop it and simply maximize

$$\sum_i \mathbb{E}_x \left(\log(\phi'((\boldsymbol{b} \cdot \boldsymbol{x})_i)) \right) + \log(\det(\boldsymbol{b})). \tag{60}$$

At this point, we may notice that that is this equation has the same form as the one which one must maximize in the ICA model where the probability density functions are replaced by the derivatives of the activation function ϕ'. This means that if one choses the cumulative distribution functions of the densities p_i's of the Independent Components as the activation functions phi, then we have that infomax is equivalent to the maximum likelihood estimation of the ICA model. In particular, using the sigmoid activation function

$$\phi(x) = \frac{1}{1 + e^{-x}}, \tag{61}$$

then ϕ' is a sparse distribution, and in fact $\log(\phi')$ is the commonly used measure of sparsity, the log-cosh function. This ties together ICA, sparse coding, and infomax.