Information Theory for Machine Learning

Massimiliano Tomassoli
(reverse(*5102mnhuik*)@gmail.com)

05/22/16

Contents

1 Introduction

Students of *Machine Learning* are usually introduced to *Information Theory* through brief tutorials which are too superficial to really understand what's going on. One could always read a specialized book, but this might prove too much of an effort for someone who doesn't intend to become an expert in Information Theory.

This tutorial wants to be a *reasoned* exposition where everything follows logically from what precedes it. I hope I succeeded in that.

Because of the intended audience, some theorems are presented without proof but are thoroughly explained to give a solid intuition of why they're true.

To make this tutorial self-contained, I took the liberty of proving some classical results such as the *Inclusion Exclusion Principle* and *Jensen's Inequality* as I needed them to prove other results.

Moreover, I decided to conclude this tutorial with two sections about the celebrated *EM Algorithm* which I think everyone should know even if they're just interested in, say, *Deep Learning*. I didn't throw in the EM Algorithm just for the fun of it: there's an interesting, if not strong, connection with Information Theory.

A **warning**: every derivation is my own so keep your eyes open and let me know if you find any mistakes!

Part I
The theory

2 Mean or Expected Value

Before we dive into *Information Theory*, we'd better review the definition and some important properties of the *mean* or *expected value*. We'll be focusing on the case of *finite discrete* random variables here, i.e. random variables which take only a finite number of values. For the *infinite* case there are some subtleties about convergence. Also, for the *continuous* case one just have to replace all the sums with integrals.

Definition 1. Let X be a discrete random variable with distribution p. The mean of X is defined as

$$\mathbb{E}_{X\sim p}[X] = \sum_x p(x)x.$$

Remark 1. When there is no ambiguity, we can drop the distribution, the variable, or both:

$$\mathbb{E}_{X\sim p}[X] = \mathbb{E}_X[X] = \mathbb{E}_p[X] = \mathbb{E}[X].$$

Definition 2. With a slight abuse of notation, if X is a random variable, we also see X as the set of values which X can take so that we can write, for instance, $\forall x \in X,\ p(x) \in [0, 1]$.

Definition 3. If X is a random variable and f a function, then $Z = f(X)$ is a random variable such that, for all z,

$$p(z) = P(Z = z) = P(f(X) = z) = P(X \in f^{-1}(z)) = \sum_{x:f(x)=z} p(x).$$

Proposition 1. *Let X be a random variable of distribution p and let f be a function. The mean of the random variable $f(X)$ can be evaluated as follows*

$$\mathbb{E}_{f(X)}[f(X)] = \mathbb{E}_X[f(X)] = \sum_x p(x)f(x).$$

Proof. Let $Z = f(X)$ and let q be the distribution of Z. Then,

$$\begin{aligned}
\mathbb{E}_{f(X)}[f(X)] &= \mathbb{E}_Z[Z] \\
&= \sum_z q(z)z \\
&= \sum_z \left(\sum_{x:f(x)=z} p(x) \right) z \\
&= \sum_z \sum_{x:f(x)=z} p(x)z \\
&= \sum_z \sum_{x:f(x)=z} p(x)f(x) \\
&= \sum_x p(x)f(x) = \mathbb{E}_X[f(X)].
\end{aligned}$$

Note that the double sum can be replaced with a single sum over x because $\{x_1, x_2, \ldots, x_n\} = \bigcup_z \{x | f(x) = z\}$. □

Definition 4. We can generalize definition 1 by considering a list of random variables $X_1, \ldots, X_n$:

$$\mathbb{E}_{X_1,\ldots X_n}[f(X_1, \ldots, X_n)] = \sum_{x_1} \cdots \sum_{x_n} p(x_1, \ldots, x_n) f(x_1, \ldots, x_n).$$

Definition 5. If $X_1, \ldots, X_n$ are random variables, we can define a corresponding **vector** random variable as

$$X = (X_1, \ldots, X_n) = [X_1 \cdots X_n]^t.$$

Proposition 2. *If $X = (X_1, \ldots, X_n)$ and $Y = (Y_1, \ldots, Y_m)$ are vector random variables, i.e. vectors of random variables, then*

$$\begin{aligned}
p(\ldots, x, \ldots) &= p(\ldots, x_1, \ldots, x_n, \ldots) \\
p(\ldots, x, \ldots | \ldots, y, \ldots) &= p(\ldots, x_1, \ldots, x_n, \ldots | \ldots, y_1, \ldots, y_m, \ldots),
\end{aligned}$$

where $x = (x_1, \ldots, x_n)$ and $y = (y_1, \ldots, y_n)$.

Proof. Let's try not to be too technical. We'll indicate *events* by writing predicates between *curly braces*. For instance, the event "X takes value x" is written as $\{X = x\}$. Now note that

$$p(x, y) = p(X = x, Y = y) = P(\{X = x\} \cap \{Y = y\}),$$

which means that $p(x, y)$ is really the probability of the intersection of the two events $\{X = x\}$ and $\{Y = y\}$. Therefore, since

$$X = x \iff X_i = x_i,\ i = 1, \ldots, n,$$

then

$$\{X = x\} = \{X_1 = x_1\} \cap \cdots \cap \{X_n = x_n\},$$

which means that

$$\begin{aligned} p(\ldots, x, \ldots) &= P(\cdots \cap \{X = x\} \cap \cdots) \\ &= P(\cdots \cap \{X_1 = x_1\} \cap \cdots \cap \{X_n = x_n\} \cap \cdots) \\ &= P(\ldots, x_1, \ldots, x_n, \ldots). \end{aligned}$$

So, basically, a *comma* in these expressions means "*and*" or "*intersection*". The proof for $p(\ldots, x, \ldots \mid \ldots, y, \ldots)$ is analogous. □

Remark 2. Thanks to proposition 2, we can always simplify notation by writing just "X" or "x" instead of "$X_1, \ldots, X_n$" or "$x_1, \ldots, x_n$", respectively, and we can generalize results by doing the opposite. For instance,

$$\mathbb{E}_X[f(X)] = \sum_x p(x) f(x)$$

implies the generalization

$$\mathbb{E}_{X_1,\ldots,X_n}[f(X_1, \ldots, X_n)] = \sum_{x_1} \cdots \sum_{x_n} p(x_1, \ldots, x_n) f(x_1, \ldots, x_n).$$

Note that, to be precise, the f in the generalization is not exactly the same as the f in the simple case, but we won't be afraid of abusing notation when convenient.

Definition 6. The mean of a vector random variable $X = (X_1, \ldots, X_n)$ is defined as

$$\mathbb{E}[X] = (\mathbb{E}[X_1], \ldots, \mathbb{E}[X_n]).$$

In general, if M is an $m \times n$ matrix random variable, then

$$[\mathbb{E}[M]]_{ij} = \mathbb{E}[M_{ij}] \quad i = 1, \ldots, m,\ j = 1, \ldots, n.$$

Proposition 3. *The mean of a constant is equal to the constant itself.*

Proof. A constant c can be seen as a random variable X which takes the value c with probability 1. Thus,

$$\mathbb{E}[c] = \mathbb{E}_X[X] = \sum_x p(x) x = c.$$

□

Proposition 4. $\mathbb{E}[\cdot]$ *is linear.*

Proof. If X is a random variable, a, b two constants, and $f(x) = a + bx$, then

$$\begin{aligned}\mathbb{E}_X[a + bX] &= \mathbb{E}_X[f(X)] \\ &= \sum_x p(x)f(x) \\ &= \sum_x p(x)(a + bx) \\ &= a\sum_x p(x) + b\sum_x p(x)x \\ &= a + b\mathbb{E}_X[X].\end{aligned}$$

□

Proposition 5. *Let X, Y be two random variables. Then*

$$\mathbb{E}_{X,Y}[f(X)] = \mathbb{E}_X[f(X)].$$

In general, we can drop any random variable the argument to the mean doesn't depend on.

Proof. This is easy:

$$\begin{aligned}\mathbb{E}_{X,Y}[f(X)] &= \sum_x \sum_y p(x, y)f(x) \\ &= \sum_x \sum_y p(x)p(y|x)f(x) \\ &= \sum_x p(x)f(x) \sum_y p(y|x) \\ &= \sum_x p(x)f(x) = \mathbb{E}_X[f(X)].\end{aligned}$$

For the general case, we can consider an arbitrary sequence $X_1, \ldots, X_n$ of random variables and prove that we can drop the first variable which doesn't appear in the argument to the mean. Since n is finite, by repeating this process we must be left with a sequence $X_{i_1}, X_{i_2}, \ldots X_{i_k}$ of random variables which all appear in the argument to the mean. □

Remark 3. Thanks to property 5, we can adopt the convention that if $X_1, \ldots, X_n$ are random variables and f is a function which depends on all of them and them alone, then we can write

$$\mathbb{E}[f(X_1, \ldots X_n)] = \mathbb{E}_{X_1,\ldots,X_n}[f(X_1, \ldots, X_n)].$$

Remark 4. Given the notation and the conventions we established, we can easily conclude that, if L is linear, then $\mathbb{E}[\cdot] \circ L = L \circ \mathbb{E}[\cdot]$. Indeed,

$$\begin{aligned}(\mathbb{E}[\cdot] \circ L)(X) = \mathbb{E}[L(X)] &= \sum_x p(x)L(x) \\ &= \sum_x L(p(x)x) \\ &= L\left(\sum_x p(x)x\right) = L(\mathbb{E}[X]) = (L \circ \mathbb{E}[\cdot])(X).\end{aligned}$$

For instance, if $L(X_1, \ldots, X_n) = \sum_{i=1}^n a_i X_i$, where $X_1, \ldots, X_n$ are random variables and $a_1, \ldots, a_n$ constants, then

$$\begin{aligned}(\mathbb{E}[\cdot] \circ L)(X_1, \ldots, X_n) &= \mathbb{E}\left[\sum_{i=1}^n a_i X_i\right] \\ &= \sum_{i=1}^n a_i \mathbb{E}[X_i] = (L \circ \mathbb{E}[\cdot])(X_1, \ldots, X_n),\end{aligned}$$

by defining $\mathbb{E}[\cdot](X_1, \ldots, X_n) = (\mathbb{E}[X_1], \ldots, \mathbb{E}[X_n])$.

We can clean things up by using vectors. If $X{=}[X_1, \ldots, X_n]^T$ and $a = [a_1, \ldots, a_n]^T$, then our example becomes

$$\begin{aligned}(\mathbb{E}[\cdot] \circ L)(X) &= \mathbb{E}[a^T X] \\ &= a^T \mathbb{E}[X] = (L \circ \mathbb{E}[\cdot])(X).\end{aligned}$$

Definition 7. Let X, Y be two random variables and f a function. The *conditional* mean of $f(X, y)$ with respect to X given Y is defined as

$$\mathbb{E}_{X|Y}[f(X, y)] = \sum_x p(x|y) f(x, y),$$

where y is any fixed real number and not a random variable, so we could define a new random variable as $Z = \mathbb{E}_{X|Y}[f(X, Y)]$. Note that we didn't write y, but Y this time. A more explicit way to write it would be $Z = \mathbb{E}_{X|Y}[f(X, \cdot)](Y)$, which makes it clear that we're transforming the variable Y through the function $y \mapsto \mathbb{E}_{X|Y}[f(X, y)]$.

Proposition 6. *If X, Y are two random variables, then*

$$\mathbb{E}_{X,Y}[f(X, Y)] = \mathbb{E}_Y\left[\mathbb{E}_{X|Y}[f(X, Y)]\right].$$

Proof. The proof is easy:

$$\begin{aligned}\mathbb{E}_{X,Y}[f(X,Y)] &= \sum_x \sum_y p(x,y) f(x,y) \\ &= \sum_x \sum_y p(x|y) p(y) f(x,y) \\ &= \sum_y p(y) \sum_x p(x|y) f(x,y) \\ &= \mathbb{E}_Y \left[\mathbb{E}_{X|Y}[f(X,Y)] \right].\end{aligned}$$

□

Proposition 7. *If X, Y, C are random variables, then, for any fixed c,*

$$\mathbb{E}_{X,Y|C}[f(X,Y,c)] = \mathbb{E}_{Y|C} \left[\mathbb{E}_{X|Y,C}[f(X,Y,c)] \right].$$

This is just a generalization of proposition 6.

Proof. This proof can be derived from the proof of proposition 6 just by adding "$|C$", "$|c$" and "$, c$" in the right places:

$$\begin{aligned}\mathbb{E}_{X,Y|C}[f(X,Y,c)] &= \sum_x \sum_y p(x,y|c) f(x,y,c) \\ &= \sum_x \sum_y p(x|y,c) p(y|c) f(x,y,c) \\ &= \sum_y p(y|c) \sum_x p(x|y,c) f(x,y,c) \\ &= \mathbb{E}_{Y|C} \left[\mathbb{E}_{X|Y,C}[f(X,Y,c)] \right].\end{aligned}$$

□

Corollary 1. *If X, Y are independent random variables, then*

$$\mathbb{E}_{X,Y}[XY] = \mathbb{E}_X[X] \mathbb{E}_Y[Y].$$

Proof. First of all, because X, Y are independent, in general, for any fixed y,

$$\begin{aligned}\mathbb{E}_{X|Y}[f(X,y)] &= \sum_x p(x|y) f(x,y) \\ &= \sum_x p(x) f(x,y) \\ &= \mathbb{E}_X[f(X,y)].\end{aligned}$$

Now we can conclude the proof:

$$\begin{aligned}\mathbb{E}_{X,Y}[XY] &= \mathbb{E}_Y \left[\mathbb{E}_{X|Y}[XY] \right] \\ &= \mathbb{E}_Y \left[\mathbb{E}_X[XY] \right] \\ &= \mathbb{E}_Y \left[Y \mathbb{E}_X[X] \right] \\ &= \mathbb{E}_X[X] \mathbb{E}_Y[Y].\end{aligned}$$

□

3 Entropy

Entropy is a function which assigns single real numbers to *finite discrete distributions*. Some people refer directly to the distributions and others to the random variables associated with them. We'll be using whatever notation is more convenient for what we want to say. *Be flexible and learn not to be thrown off by different notations!*

In what follows, X, Y, Z are always *discrete* random variables, p, q, r are *finite discrete* distributions, and x, y, z are values taken by the variables X, Y, Z, respectively.

Definition 8. Self-information

If X follows distribution p, the *self-information* of x is defined as

$$I(x) = \log_2 \frac{1}{p(x)} = -\log_2 p(x).$$

The formula above says that values with high probability have low self-information, whereas values with low probability have high self-information. We'll see later why this makes sense.

Note that I measures the *amount* of information. We'll often say just "information" instead of "amount of information".

Definition 9. Entropy

The *entropy* of X is defined as

$$H(X) = \mathbb{E}_X[I(X)] = -\sum_x p(x) \log_2 p(x).$$

The entropy is the *expected* amount of information contained in a random variable. A variable may assume values $x_1, x_2, \ldots, x_n$ and each value may have a different self-information so it makes sense to summarize the information contained in all the values with the mean.

As always, note that we didn't write $\mathbb{E}_X[I(x)]$ with a lowercase x. While $I(x)$ is just a real number, $I(X)$ is a random variable obtained by applying the function I to the random variable X.

4 Entropy and Codes

Consider the following scenario. We have a random variable X with distribution p. We take samples $x_1, x_2, \ldots$ from p and send these values through a channel. We send the samples as we generate them so this process may go on forever.

To send the samples, we must encode them some way. We can define a *code*, i.e. a function $C : X \to \Sigma^*$ which maps values x to *words* over an alphabet Σ, which is just a set of arbitrary *symbols*. The asterisk is the *Kleene star*, which takes a set of symbols and returns the set of all the words (of finite non-negative length) over those symbols. For instance,

$$\{0, 1\}^* = \{\epsilon, 0, 1, 00, 01, 10, 11, 000, 001, \ldots\}$$

is the set of all the finite binary strings, including the empty string (denoted by ϵ).

The *extension* of C is defined as follows:

$$C^*(x_1, x_2, \ldots, x_n) = C(x_1)C(x_2)\cdots C(x_n)$$

where the juxtaposition indicates *concatenation* of strings. Basically, we encode sequences of samples x_i by concatenating the encoded samples. We say that C is *uniquely decodable* if C^* is *injective*. Note that if C^* is not injective, then there are two different sequences of samples which result in the same encoded string and thus, given the encoded string, we can't recover the original sequence of samples (we can recover both sequences but which one is the right one?).

Theorem 1. *Shannon's Source Code Theorem*

Let X be a discrete random variable and $C : X \to \Sigma^*$ a uniquely decodable code. If C is optimal, i.e. it minimizes $\mathbb{E}_X[\text{length}(C(X))]$, then

$$\frac{H(X)}{\log_2 |\Sigma|} \leq \mathbb{E}_X[\text{length}(C(X))] < \frac{H(X)}{\log_2 |\Sigma|} + 1.$$

To understand theorem 1 better, note that if we choose $\Sigma = \{0, 1\}$, i.e. we encode the symbols as binary strings, then the inequality simplifies to

$$H(X) \leq \mathbb{E}_X[\text{length}(C(X))] < H(X) + 1.$$

We can also be more explicit:

$$H(X) \leq \min_C \mathbb{E}_X[\text{length}(C(X))] < H(X) + 1.$$

So, we can't do better than $H(X)$: no matter which code we choose, the expected length of the encoded symbols will be at least $H(X)$. Note, though, that if the code is optimal then we can't do worse than $H(X) + 1$. But where does that $+1$ come from? Before finding that out, we must observe a few things.

You should remember that $H(X)$ is the expected self-information, so we can rewrite the formula above as

$$\mathbb{E}_X[I(X)] \leq \min_C \mathbb{E}_X[\text{length}(C(X))] < \mathbb{E}_X[I(X)] + 1.$$

This suggests that $I(X)$ has something to do with $\text{length}(C(X))$. In fact, $I(x)$ measures the information "contained" in x by counting the number of *bits* an optimal code uses to represent x. As we observed before, the definition $I(x) = \log_2 \frac{1}{p(x)}$ suggests that an optimal code will use short strings for frequent samples and long strings for rare samples. (As a side note, observe that if X was a *continuous* random variable then any sample x would be infinitely rare and our reasoning would fall apart!) This explains, at least in part, why the self-information and the entropy contain a logarithm.

If we choose a code which assigns each value x a string whose length is given by $I(x)$, then the expected length of the encoded samples is $H(X)$. Unfortunately, $I(x)$ is not always an integer, so the best we can do is take the smallest integer greater than or equal to $I(x)$, which is $\lceil I(x) \rceil$. We have the following:

$$\begin{aligned} I(x) \leq \lceil I(x) \rceil < I(x) + 1 \\ \mathbb{E}_X[I(X)] \leq \mathbb{E}_X \lceil I(X) \rceil < \mathbb{E}_X[I(X) + 1] = \mathbb{E}_X[I(X)] + 1 \\ H(X) \leq \mathbb{E}_X \lceil I(X) \rceil < H(X) + 1. \end{aligned} \tag{1}$$

For this derivation to make sense, we should also prove that there exists an actual (uniquely decodable) code C such that $\text{length}(C(x)) = \lceil I(x) \rceil$, but, lucky for us, our main goal is just to understand things intuitively!

4.1 Can we do better?

We saw that by encoding one sample at a time we may waste up to one bit per sample on average since we need to round $I(x)$ up to the nearest integer. What would happen if we grouped samples in blocks of n samples and encoded a group at a time?

The easiest way to do this is to define an n-dimensional random variable $Z = (X_1, X_2, \ldots, X_n)$, where the X_i are independent copies of X. This makes sense because taking n samples from the same variable X is the same as taking a single sample from each one of n i.i.d (independent and identically distributed) variables X_i which follows the same distribution as X.

Because $X_1, X_2, \ldots, X_n$ are independent, $p(z) = p(x_1)p(x_2) \cdots p(x_n)$. This means that

$$\begin{aligned} I(z) &= -\log_2 p(z) \\ &= -\log_2 \prod_{i=1}^{n} p(x_i) \\ &= -\sum_{i=1}^{n} \log_2 p(x_i) \\ &= \sum_{i=1}^{n} I(x_i). \end{aligned}$$

As a consequence,

$$\begin{aligned} H(Z) &= \mathbb{E}_Z[I(Z)] \\ &= \mathbb{E}_Z\left[\sum_{i=1}^{n} I(X_i)\right] \\ &= \sum_{i=1}^{n} \mathbb{E}_{X_i}[I(X_i)] \\ &= \sum_{i=1}^{n} H(X_i) \\ &= nH(X) \end{aligned}$$

thanks to the linearity of $\mathbb{E}[\cdot]$ and the fact that the X_i are independent copies of X.

Thanks to inequality 1, we get the following:

$$\begin{aligned} H(Z) \leq\ & \mathbb{E}_Z\lceil I(Z)\rceil\ < H(Z)+1 \quad \iff \\ nH(X) \leq\ & \mathbb{E}_Z\lceil I(Z)\rceil\ < nH(X)+1 \iff \\ H(X) \leq\ & \frac{\mathbb{E}_Z\lceil I(Z)\rceil}{n}\ < H(X)+\frac{1}{n}. \end{aligned}$$

Since $\mathbb{E}_Z\lceil I(Z)\rceil$ is the average number of bits sent for one block of n samples, by dividing by n we get the average number of bits sent per sample.

As we can readily see, by encoding n samples at a time, we've narrowed the interval by n times. This means that, at least in theory, we can get as close to $H(X)$ as we like. Equivalently, $\frac{\mathbb{E}_Z\lceil I(Z)\rceil}{n} \to H(X)$ as $n \to +\infty$. This gives us another version of theorem 1:

Theorem 2. *Shannon's Source Code Theorem (asymptotic version)*

A sequence of n samples generated from a discrete random variable X with entropy $H(X)$ can be compressed into $nH(X)$ bits on average with negligible loss as $n \to \infty$. Conversely, no uniquely decodable code can do better without incurring loss of information.

Note that if n samples can be compressed into $nH(X)$ bits then each sample is compressed into $H(X)$ on average, thus the theorem is saying the same thing. Also, some authors prefer to talk about "n i.i.d. random variables $X_1, X_2, \ldots, X_n$" instead of "n samples generated from a random variable X". As we said before, they're just two different ways of saying the same thing.

5 Logarithm and probabilities

Is there a way to justify the logarithm in the definition of the self-information and, as a consequence, of the entropy without having to resort to codes and lengths of binary strings?

We know that $H(X)$ wants to be a measure of the (amount of) information that we acquire, on average, when we observe the value of the random variable X. For instance, let's take $X \sim Ber(\mu)$, which means that $X = \{0, 1\}$ and

$$p(x) = \begin{cases} \mu & \text{if } x = 1 \\ 1 - \mu & \text{if } x = 0 \end{cases}$$

or, more concisely, $p(x) = \mu^x (1 - \mu)^{1-x}$. To generate samples from X we could use a coin which lands heads with probability μ.

If we flip the coin and observe the outcome, we have acquired, on average, $H(X)$ bits of information. Now let's say we flip the coin again and observe the outcome. What's the total amount of information we have acquired? Maybe $H(X)^2$? Or maybe $2H(X)$? The latter makes more sense because each toss gives us, on average, the same amount of information and the two tosses are independent so they don't interact in any way and adding them up seems the more reasonable thing to do.

Now let's consider $Z = (X_1, X_2)$ where $X_1 \sim Ber(\mu)$, $X_2 \sim Ber(\mu)$ and $p(x_1, x_2) = p(x_1)p(x_2)$, i.e. the two random variables are independent. Observing a sample from Z should give us, on average, the same amount of information as observing, together, a sample from X_1 and a sample from X_2.

We can make things even simpler and reason about I. Let's assume that we flip the coin twice and we get the values x_1, x_2. The amount of information we acquire is $I(z) = I(x_1) + I(x_2)$, where $z = (x_1, x_2)$ is the equivalent sample from Z.

Another important assumption we must make is that the self-information of each sample only depends on its probability, i.e. $I(x) = f(p(x))$ for some function f. In other words, $I(x_1) = I(x_2) \iff p(x_1) = p(x_2)$.

Here's what we can conclude about f for now:

$$\begin{aligned} I(z) = I(x_1) + I(x_2) \quad &\iff \\ f(p(z)) = f(p(x_1)) + f(p(x_2)) \quad &\iff \\ f(p(x_1)p(x_2)) = f(p(x_1)) + f(p(x_2)). \end{aligned}$$

The values $p(x_1)$ and $p(x_2)$ are not completely arbitrary ($p(x_1), p(x_2) \in [0, 1]$ and $p(x_1) + p(x_2) \leq 1$) but, for simplicity, let's pretend they are. We can conclude that

$$\forall x, y \in \mathbb{R}, \quad f(xy) = f(x) + f(y).$$

Another reasonable assumption is that $I(1) = 0$ because if, for instance, a coin lands heads with probability 1, then tossing the coin and observing the outcome (heads) doesn't tell us anything we didn't already know. This means that $f(1) = 0$.

We should also assume that f is *strictly decreasing*, i.e.

$$x < y \implies f(x) > f(y)$$

because the rarer and thus more surprising the event, the higher the amount of information acquired by observing that event.

It turns out that the logarithm is the only *strictly increasing* function $g : (0, +\infty) \to \mathbb{R}$ such that $g(1) = 0$ and $\forall x, y \in \mathbb{R},\ \ g(xy) = g(x) + g(y)$. Since $-f$ is strictly increasing, this means that

$$\begin{aligned} -f(x) &= \log_b x \qquad \iff \\ f(x) &= -\log_b x \qquad \implies \\ I(x) &= -\log_b p(x). \end{aligned}$$

We can choose $b = 2$ which corresponds to measuring the amount of information in bits.

Intuitively, we can say that the logarithm makes sense because *probabilities multiply while amounts of information add up* and what better way to transform products into sums than using the logarithm?

6 Jensen's Inequality

Jensen's Inequality is a very useful and powerful tool. It has to do with *convex* and *concave* functions so let's talk about *convexity* first.

6.1 Convexity

Definition 10. If $x_1, \ldots, x_n$ are some points in a vector space (e.g. $\mathbb{R}^n$), then a *convex combination* of them is any point x_α defined as follows:

$$x_\alpha = \sum_{i=1}^{n} \alpha_i x_i, \ \ \alpha > 0, \sum_{i=1}^{n} \alpha_i = 1,$$

where $\alpha = (\alpha_1, \ldots, \alpha_n)$ and $\alpha > 0 \iff (\alpha_i > 0,\ i = 1, \ldots, n)$. Note that α can be interpreted as a *finite discrete distribution*.

Definition 11. A set is *convex* if and only if any convex combination of any of its points is in the set.

Proposition 8. *A set is* convex *if and only if any convex combination of any pair of its points is in the set. See picture 1.*

Proof. Let's call a convex combination of n points an n-combination. Basically, we want to prove that definition 11 doesn't need to consider n-combinations with $n > 2$. To do this, it's enough to prove that any n-combination can be seen as a sequence of 2-combinations.

We can prove this by *induction*. The case for $n = 2$ is trivial, and assuming this is true for n, we can prove this is also true for $n + 1$:

$$\sum_{i=1}^{n+1} \alpha_i x_i = (1 - \alpha_{n+1}) \left[\sum_{i=1}^{n} \frac{\alpha_i}{1 - \alpha_{n+1}} x_i \right] + \alpha_{n+1} x_{n+1}.$$

□

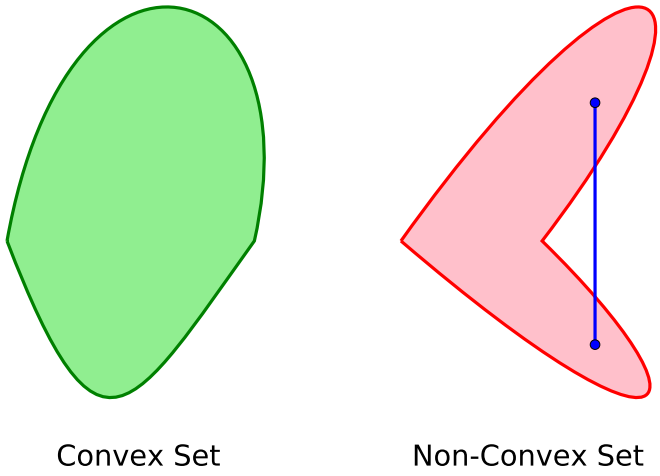


Figure 1: Example of *convex* and *non*-convex sets.

Definition 12. A function is convex if and only if the *region* above its graph is a convex set. See picture 2 for the unidimensional case.

Proposition 9. *A function is convex if and only if all the planes tangent to it are completely below it. See picture 4 for an example.*

Definition 13. A function f is *concave* if and only if $-f$ is *convex*.

6.2 Definition and proof of Jensen's Inequality

We know that $\text{Var}[X] \geq 0$ for any random variable X. We can use this fact to prove a particular case of Jensen's Inequality:

$$\begin{aligned} 0 \leq \text{Var}[X] &= \mathbb{E}\left[(X - \mathbb{E}[X])^2\right] \\ &= \mathbb{E}\left[X^2 - 2X\mathbb{E}[X] + \mathbb{E}[X]^2\right] \\ &= \mathbb{E}\left[X^2\right] - 2\mathbb{E}[X]\mathbb{E}[X] + \mathbb{E}[X]^2 \\ &= \mathbb{E}\left[X^2\right] - \mathbb{E}[X]^2 \end{aligned}$$

which implies that

$$\mathbb{E}\left[X^2\right] \geq \mathbb{E}[X]^2.$$

Note that $f(x) = x^2$ is a convex function. In general, we have the following result.

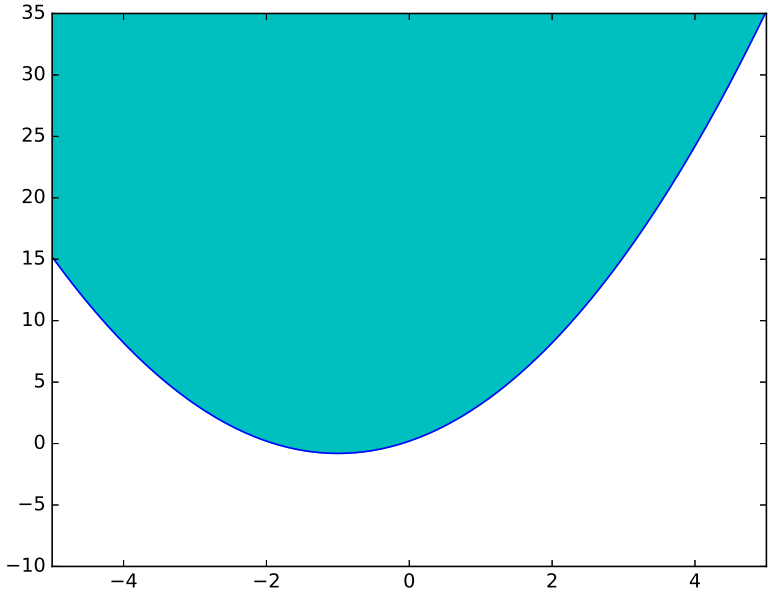


Figure 2: Example of a *convex* function. Note that the region above the graph is a *convex set* (even if *unbounded*).

Proposition 10. *(Jensen's Inequality) If f is a convex function and X a random variable, then $\mathbb{E}[f(X)] \geq f(\mathbb{E}[X])$. The* equality *holds* if and only if *f is linear or X is constant.*

Proof. For an intuitive understanding of why the inequality holds, see picture 3.

Let's consider the unidimensional case first. Let $f : \mathbb{R} \to \mathbb{R}$ be a convex function. Let's define a function $L(x) = ax+b$, where a and b are real constants, such that $L(x) \leq f(x)$ for all $x \in \mathbb{R}$ and $L(\mathbb{E}[X]) = f(\mathbb{E}[X])$. See figure 4.

Since $f(x) \geq L(x)$ for all $x \in \mathbb{R}$, we can take the mean of both sides:

$$\begin{aligned}\mathbb{E}[f(X)] &\geq \mathbb{E}[L(X)] \\ &= \mathbb{E}[aX + b] \\ &= a\mathbb{E}[X] + b \\ &= L(\mathbb{E}[X]) \\ &= f(\mathbb{E}[X]).\end{aligned}$$

If X is constant with $p(x_0) = 1$, then $f(x_0) = \mathbb{E}[f(X)] \geq f(\mathbb{E}[X]) = f(x_0)$, therefore the equality holds. Also, if f is linear, then $\mathbb{E}$ and f, both linear, can be exchanged (see remark 4) and the equality holds again.

It's easy to generalize the result to the case of $f : \mathbb{R}^n \to \mathbb{R}$ by defining

$L(x) = a^T x + b$, where a is a fixed n-dimensional vector and b is a fixed scalar:

$$\begin{aligned}
\mathbb{E}[f(X)] &\geq \mathbb{E}[L(X)] \\
&= \mathbb{E}[a^T X + b] \\
&= \mathbb{E}\left[\sum_{i=1}^{n} a_i X_i + b\right] \\
&= \sum_{i=1}^{n} a_i \mathbb{E}[X_i] + b \\
&= a^T \mathbb{E}[X] + b \\
&= L(\mathbb{E}[X]) \\
&= f(\mathbb{E}[X]).
\end{aligned}$$

□

7 Maximum Entropy

We've seen that $I(x) = 0$ when $p(x) = 1$. This means that if X can take only one value x_0 then $H(X) = \mathbb{E}_X[I(X)] = I(x_0) = -\log_2 p(x_0) = -\log_2 1 = 0$. But when is $H(X)$ maximum?

Let X be a random variable which takes values $x_1, x_2, \ldots, x_n$ with probabilities $p_1, p_2, \ldots, p_n$, respectively. Then

$$H(X) = \mathbb{E}_X[I(X)] = -\sum_{i=1}^{n} p_i \log_2 p_i.$$

We want to find

$$\underset{p_1,p_2,\ldots,p_n}{\operatorname{argmax}} \; -\sum_{i=1}^{n} p_i \log_2 p_i.$$

Since log is concave (i.e. $-log$ is convex) we can use Jensen's Inequality:

$$\begin{aligned}
H(X) &= \mathbb{E}\left[\log_2 \frac{1}{p(X)}\right] \\
&\leq \log_2 \mathbb{E}\left[\frac{1}{p(X)}\right] \\
&= \log_2 \sum_{i=1}^{n} p_i \frac{1}{p_i} \\
&= \log_2 n.
\end{aligned}$$

Also, the equality holds when the random variable, i.e. $\frac{1}{p(X)}$, is constant, which means that $p_1 = p_2 = \ldots = p_n = \frac{1}{n}$. Note that we used proposition 1 in the derivation above.

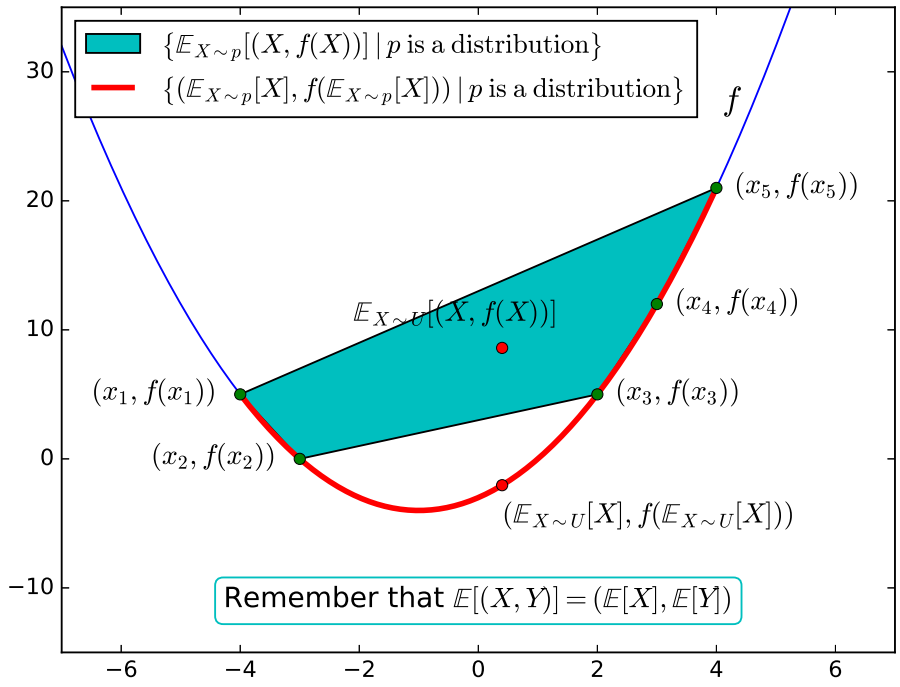


Figure 3: This figure shows Jensen's Inequality in action.
First of all, note that f is *convex*. In this example, all the distributions are defined over the five points $(x_i, f(x_i))$, $i = 1, \ldots, 5$. In particular, U is the *Uniform* distribution.
Note that $v = \mathbb{E}_{X\sim U}[(X, f(X))] = (\mathbb{E}_{X\sim U}[X], \mathbb{E}_{X\sim U}[f(X)])$ is a *convex combination* of the five points and, thus, v must be in the *convex set* shown in the figure. (We chose $p = U$, but this is true for *all* distributions.)
Since $w = (\mathbb{E}_{X\sim U}[X], f(\mathbb{E}_{X\sim U}[X]))$ must be on the *red* curve (a portion of f) and $v_x = w_x$, it follows that $v_y \geq w_y$, i.e. $\mathbb{E}_{X\sim U}[f(X)] \geq f(\mathbb{E}_{X\sim U}[X])$.
Now observe that if X is a *constant* then the five points are all equal, i.e. they become a single point. As a consequence, the convex set and the red portion of f collapse into a single point and the equality holds.
Also, if f is *linear* than its graph is a straight line and, thus, the red portion of f coincides with the convex set. Therefore, the equality holds once again.

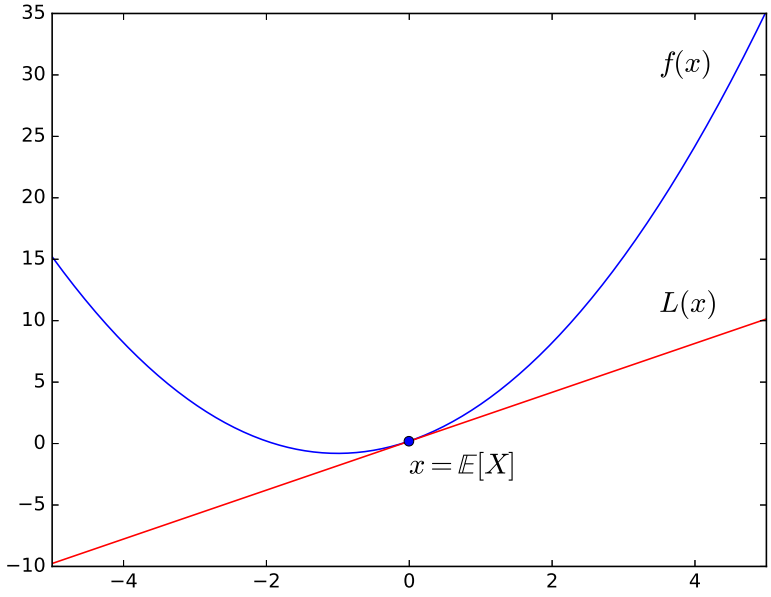


Figure 4: $L(x) \leq f(x)$ for all $x \in \mathbb{R}$ and $L(\mathbb{E}[X]) = f(\mathbb{E}[X])$.

8 Specifying the distribution

Sometimes it's useful or even necessary to indicate the distribution of a variable explicitly. Let X be a discrete random variable and p a distribution for X. Then we may write:

$$I(x) = I_p(x) = -\log_2 p(x)$$
$$H(X) = H(p) = \mathbb{E}_{X\sim p}[I(X)] = -\sum_x p(x)\log_2 p(x).$$

9 Chain rule

The *chain rule* for the entropy is very simple and shouldn't surprise you. It derives directly from what we could call the *self-information chain rule*:

$$I(x, y) = I(x|y) + I(y) = I(y|x) + I(x). \tag{2}$$

This derives directly from

$$p(x, y) = p(x|y)p(y) = p(y|x)p(x). \tag{3}$$

In fact, by taking the logarithm and negating 3 we get

$$-\log_2 p(x, y) = -\log_2 p(x|y) - \log_2 p(y) = -\log_2 p(y|x) - \log_2 p(x)$$
$$I(x, y) = I(x|y) + I(y) = I(y|x) + I(x).$$

Now we can take the mean of equality 2:

$$\begin{aligned}\mathbb{E}_{X,Y}[I(X,Y)] &= \mathbb{E}_{X,Y}[I(X|Y)] + \mathbb{E}_{X,Y}[I(Y)] \\ &= \mathbb{E}_{X,Y}[I(Y|X)] + \mathbb{E}_{X,Y}[I(X)] \iff \\ H(X,Y) &= H(X|Y) + H(Y) \\ &= H(Y|X) + H(X).\end{aligned}$$

Of course, if X and Y are independent we simply have

$$\begin{aligned}I(x,y) &= I(x) + I(y) \\ H(X,Y) &= H(X) + H(Y).\end{aligned}$$

$H(X,Y)$ is usually called the *joint entropy* of X, Y, while $H(X|Y)$ is called the *conditional entropy*. Note that we can define the latter in two ways:

$$H(X|Y) = \mathbb{E}_{X,Y}[I(X|Y)] = \mathbb{E}_Y[H(X|Y)] = \sum_y p(y)H(X|y).$$

Pay particular attention to the term $H(X|y)$ and note that the y is lowercase. This is short for $H(X|Y = y)$ which means that Y is fixed and equal to y. Also, the first $H(X|Y)$ is a scalar, while the one inside $\mathbb{E}_Y[\cdot]$ is a random variable obtained by transforming Y through the function $y \mapsto H(X|y)$.

In accordance with the product rule for the mean (see proposition 6), we must have

$$H(X|y) = \mathbb{E}_{X|Y}[I(X|y)] = -\sum_x p(x|y)\log_2 p(x|y).$$

This tells us the amount of information, after we've already observed that $Y = y$, acquired on average by observing the value of X. In general, this amount depends on the particular value y observed. If we want a measure independent of y, we can take the mean with respect to Y ending up with

$$H(X|Y) = \mathbb{E}_Y[\mathbb{E}_{X|Y}[I(X|Y)]] = \mathbb{E}_{X,Y}[I(X|Y)],$$

which tells us the amount of information, after we've already observed the value of Y (whatever the value), acquired on average by observing the value of X. Again, note that this measure doesn't depend on the particular value of Y. In fact, we average over all the possible values of Y to compute a "summary".

10 Cross Entropy

The *cross entropy* of two distributions p, q for the discrete random variable X is defined as

$$H(p,q) = \mathbb{E}_{X\sim p}[I_q(X)] = -\sum_x p(x)\log_2 q(x).$$

Note that this has nothing to do with the joint entropy $H(X, Y)$!

What's the meaning of cross entropy? $I_q(x)$ is the self-information of x assuming that $X \sim q$, so what's the meaning of taking the mean with respect to p rather than to q? This time it's better to think about *codes*.

If $C_q : X \to \{0, 1\}^*$ is an optimal code for X assuming that $X \sim q$, then $\text{length}(C_q(x)) \approx I_q(x)$. Now we can rewrite the cross entropy as

$$H(p, q) = \mathbb{E}_{X \sim p}[\text{length}(C_q(X))]$$

which is the number of bits required on average to transmit each sample x from $X \sim p$, using a code optimized for the case when $X \sim q$. It's clear that

$$H(p) = H(p, p) \leq H(p, q)$$

because C_q is not the optimal code for $X \sim p$.

In other words, $H(p, q)$ is the expected number of bits (per sample) required to transmit samples from $X \sim p$ when using a code optimized for $X \sim q$. More concisely, $H(p, q)$ is the number of bits required for $X \sim p$ when optimizing for $X \sim q$. We could also choose better names for the *formal parameters* (this is programming lingo) for the cross entropy: $H(real, optimize\_for)$. That is, the transmitted values follows the $real$ distribution, but we're optimizing for the $optimize\_for$ distribution instead so, if $real \neq optimize\_for$, we lose efficiency and waste bits.

Please note that the following is false, in general:

$$H(q) = H(q, q) \leq H(p, q) \qquad \text{(THIS IS } \boldsymbol{WRONG}\text{!!!)}$$

As a counterexample, consider $p(x) \sim Ber(0.1)$ and $q(x) \sim Ber(0.2)$. We have:

$$H(q, q) = -\sum_x q(x) \log_2 q(x) = -0.2 \log_2 0.2 - 0.8 \log_2 0.8 \approx 0.722$$

$$H(p, q) = -\sum_x p(x) \log_2 q(x) = -0.1 \log_2 0.2 - 0.9 \log_2 0.8 \approx 0.522.$$

11 Kullback-Leibler Divergence

The *Kullback-Leibler Divergence (KL Divergence)* is often used to measure the distance between two distributions, but it's not a real distance. In fact, it isn't even symmetric, as we'll see.

The KL Divergence between two distributions p and q of a variable X is defined as

$$KL(p||q) = \sum_x p(x) \log_2 \frac{p(x)}{q(x)}.$$

We can rewrite this in many ways:

$$\begin{aligned} KL(p||q) &= \mathbb{E}_{X\sim p}\left[\log_2 \frac{p(X)}{q(X)}\right] \\ &= \mathbb{E}_{X\sim p}[\log_2 p(X) - \log_2 q(X)] \\ &= \mathbb{E}_{X\sim p}[I_q(X) - I_p(X)] \\ &= \mathbb{E}_{X\sim p}[I_q(X)] - \mathbb{E}_{X\sim p}[I_p(X)] \\ &= H(p,q) - H(p,p) \\ &= H(p,q) - H(p). \end{aligned}$$

The final expression, $KL(p||q) = H(p,q) - H(p)$, tells us that $KL(p||q)$ is the price we pay when we transmit samples generated from $X \sim p$ by using an optimal code for q rather than for p, measured in bits wasted. It's intuitively clear that $KL(p||q) \geq 0$, but maybe we should try to prove it.

Once again, we can use Jensen's Inequality by noticing that $-log$ is a convex function:

$$\begin{aligned} KL(p||q) &= \mathbb{E}_{X\sim p}\left[\log_2 \frac{p(X)}{q(X)}\right] \\ &= \mathbb{E}_{X\sim p}\left[-\log_2 \frac{q(X)}{p(X)}\right] \\ &\geq -\log_2 \mathbb{E}_{X\sim p}\left[\frac{q(X)}{p(X)}\right] \\ &= -\log_2 \sum_x p(x)\frac{q(x)}{p(x)} \\ &= -\log_2 1 \\ &= 0. \end{aligned}$$

Also, since $-log_2$ is not linear, the equality holds if and only if $\frac{q(X)}{p(X)}$ is a constant random variable, which requires that $q(x) = Kp(x)$ for all $x \in X$ and some constant K. By summing up, we get

$$1 = \sum_x q(x) = K \sum_x p(x) = K$$

and so $q = p$.

In conclusion, $KL(p||q) \geq 0$ and $KL(p||q) = 0 \iff p = q$.

12 Mutual Information

12.1 Definition

The *mutual information* between two random variables X and Y measures the amount of information we learn about one variable by observing the other.

Let X, Y be two independent random variables, i.e. $p(x, y) = p(x)p(y)$ for all $x \in X, y \in Y$. It seems logical to require that the mutual information between X and Y be 0, since X tells nothing about Y and vice versa.

Now consider two random variables X, Y where $Y = X$. The mutual information between X and Y should be maximum because one completely determines the other. Note that, in this case, $p(x, y) = p(x) = p(y)$ for $x = y$, and 0 otherwise.

Intuitively, the mutual information should be related to the distance between $p(x, y)$ and $p(x)p(y)$:

1. if X, Y are independent, the distance between the distributions $p(x, y)$ and $p(x)p(y)$ should be 0;
2. if $X = Y$, the distance between $p(x, y)$ and $p(x)p(y)$ should be maximum.

Let's see if we can use the KL Divergence for this. We'll use $I(X; Y)$ to denote the mutual information between X and Y, and abuse notation a little by specifying the arguments of the three $p(\cdot)$ in order to tell them apart:

$$\begin{aligned} I(X; Y) &= KL(p(x, y)||p(x)p(y)) \\ &= H(p(x, y), p(x)p(y)) - H(p(x, y)) \\ &= \sum_x \sum_y p(x, y) \log_2 \frac{p(x, y)}{p(x)p(y)}. \end{aligned}$$

In section 11 we proved that

$$KL(p||q) = 0 \iff p = q,$$

which means that $I(X; Y) = 0$ if and only if X, Y are independent, so our first requirement is met.

Now we need to check whether $I(X; Y)$ is maximum when $Y = X$. As we've already said, if $Y = X$ then

$$p(x, y) = \begin{cases} p(x) & \text{if } x = y \\ 0 & \text{otherwise.} \end{cases}$$

Therefore

$$\begin{aligned} I(X; Y) &= \sum_x \sum_y p(x, y) \log_2 \frac{p(x, y)}{p(x)p(y)} \\ &= \sum_x \sum_{y:y=x} p(x, y) \log_2 \frac{p(x, y)}{p(x)p(y)} \\ &= \sum_x p(x) \log_2 \frac{p(x)}{p(x)^2} \\ &= -\sum_x p(x) \log_2 p(x) = H(X). \end{aligned}$$

Note that $I(X;Y) = H(X) = H(Y)$ because we can repeat the derivation by keeping y instead of x. This means that X reveals, *on average*, $H(Y)$ bits of information for Y which corresponds to *all* the information contained in Y. Of course, the same is true for Y and $H(X)$. In fact, note that $I(X,Y)$ is *symmetric*.

So it seems our definition is promising; in fact, this is *exactly* how mutual information is defined in the literature.

12.2 Alternative formulation and interpretation

We saw that $I(X;Y)$ is the amount of information about one variable acquired, on average, by observing the other variable. Can't we measure that by using entropy directly?

We know that $H(X)$ is the amount of information acquired, on average, by observing X and that $H(X|Y)$ is the amount of information acquired, on average, by observing X when we have already observed Y. It's as if X contained a certain (expected) amount of information and Y revealed a portion of that information. Thus, $H(X|Y)$ is the information in X not revealed by Y. As a consequence, the expected amount of information about X revealed by Y should be $H(X) - H(X|Y)$.

Indeed,

$$\begin{aligned} I(X;Y) &= \sum_x \sum_y p(x,y) \log_2 \frac{p(x,y)}{p(x)p(y)} \\ &= \sum_x \sum_y p(x,y) \log_2 \frac{p(x|y)}{p(x)} \\ &= \mathbb{E}_{X,Y}[-\log_2 p(x)] - \mathbb{E}_{X,Y}[-\log_2 p(x|y)] \\ &= H(X) - H(X|Y). \end{aligned}$$

12.3 Mutual Information VS Correlation

Why do we need mutual information? Can't we use *correlation* instead?

The correlation between two random variables X and Y is defined as

$$\rho_{XY} = \frac{\mathrm{Cov}[X,Y]}{\sqrt{\mathrm{Var}[X]\mathrm{Var}[Y]}}.$$

The denominator is just used to normalize the correlation so that it's in $[-1,1]$.

If X and Y are independent then $\mathrm{Cov}[X,Y] = 0$. This is easy to prove. First, let's write the usual definition of the covariance and then "simplify" it:

$$\begin{aligned} \mathrm{Cov}[X,Y] &= \mathbb{E}\left[(X - \mathbb{E}[X])(Y - \mathbb{E}[Y])\right] \\ &= \mathbb{E}\left[XY - X\mathbb{E}[Y] - \mathbb{E}[X]Y + \mathbb{E}[X]\mathbb{E}[Y]\right] \\ &= \mathbb{E}[XY] - \mathbb{E}[X]\mathbb{E}[Y] - \mathbb{E}[X]\mathbb{E}[Y] + \mathbb{E}[X]\mathbb{E}[Y] \\ &= \mathbb{E}[XY] - \mathbb{E}[X]\mathbb{E}[Y]. \end{aligned}$$

By corollary 1, we know that if X and Y are independent then $\mathbb{E}[XY] = \mathbb{E}[X]\mathbb{E}[Y]$ and we must also have $\text{Cov}[X, Y] = 0$.

So independence implies zero covariance. But what about the converse? Does zero covariance implies independence like mutual information does?

The answer is no and here's a counterexample. Let X, Y be two random variables such that $X \in \{-1, 1\}$ with $p(x) = 0.5$ for both $x \in X$, and such that $Y = X^2$. Then

$$\begin{aligned}\text{Cov}[X, Y] &= \mathbb{E}[XY] - \mathbb{E}[X]\mathbb{E}[Y] \\ &= \mathbb{E}[X^3] - \mathbb{E}[X]\mathbb{E}[X^2] \\ &= 0\end{aligned}$$

because $X \sim X^3$ (i.e. they're *identically distributed*) and both have zero mean.

The problem with covariance and correlation is that they only measure *linear* dependence. In fact, in the counterexample above, Y does depend on X, but not linearly.

12.4 Linear Regression

Here's a way to see why covariance only measures linear dependence. Let X and Y be random variables.

We want to find a, b such that the *linear* function $f(x) = ax + b$ describes as closely as possible the relation between X and Y. To do this, we'll minimize the *mean squared error*:

$$L(a, b) = \mathbb{E}_{X,Y}\left[\frac{1}{2}(f(X) - Y)^2\right].$$

We can use Calculus to minimize $L(a, b)$:

$$\begin{aligned}\frac{\partial L(a,b)}{\partial a} &= \frac{\partial}{\partial a}\mathbb{E}\left[\frac{1}{2}(f(X) - Y)^2\right] \\ &= \mathbb{E}\left[\frac{1}{2}\frac{\partial}{\partial a}(f(X) - Y)^2\right] \\ &= \mathbb{E}[(f(X) - Y)X] \\ &= a\mathbb{E}\left[X^2\right] + b\mathbb{E}[X] - \mathbb{E}[XY] = 0 \qquad (4)\end{aligned}$$

$$\begin{aligned}\frac{\partial L(a,b)}{\partial b} &= \frac{\partial}{\partial b}\mathbb{E}\left[\frac{1}{2}(f(X) - Y)^2\right] \\ &= \mathbb{E}\left[\frac{1}{2}\frac{\partial}{\partial b}(f(X) - Y)^2\right] \\ &= \mathbb{E}\left[f(X) - Y\right] \\ &= a\mathbb{E}[X] + b - \mathbb{E}[Y] = 0. \qquad (5)\end{aligned}$$

By combining equations 4 and 5, we get

$$a = \frac{\mathbb{E}[XY] - \mathbb{E}[X]\mathbb{E}[Y]}{\mathbb{E}\left[X^2\right] - \mathbb{E}[X]^2} = \frac{\mathrm{Cov}[X,Y]}{\mathrm{Var}[X]}.$$

If $\mathrm{Var}[X] = 1$ the *slope* of the line that best captures the linear dependence between X and Y is exactly $\mathrm{Cov}[X,Y]$. Since the regression line isn't capable of capturing *nonlinear* dependences between X and Y, neither is the covariance nor the correlation (note that if $\mathrm{Var}[X] = \mathrm{Var}[Y] = 1$ then $a = \rho_{XY}$ so, as we said before, the correlation is just a covariance normalized).

13 A Set Theoretic view of Information Theory

During our discussion about entropy and related measures, we used some expressions such as "the information contained in [...]" which might reminds us of *Set Theory*. Let's see if we can indeed view Information Theory through the eyes of a set theorist.

Let A be a random variable. We know that $H(A)$ measures the amount of information contained in A, so maybe we could view A as a set and $H(A)$ as some kind of "*cardinality*" of A. The usual cardinality counts the number of elements in a set, while our cardinality might count the number of bits of information in the set. In agreement with our intuition, let's adopt the following notation:

$$|A| = H(A)$$

If two random variables A, B are independent, one doesn't tell anything about the other so we could say that they *share no information*, i.e. $|A \cap B| = 0$.

We saw that, in general, $H(A,B) = H(A|B) + H(B)$, but, if A and B are independent, then $H(A,B) = H(A) + H(B)$. Doesn't this remind you of the *Inclusion Exclusion Principle (IEP)*?

In its simplest form, the IEP says that

$$|S \cup T| = |S| + |T| - |S \cap T|,$$

where this time $|\cdot|$ denotes proper cardinality and S, T are two real sets. The idea behind the formula is that if S and T have elements in common, then $|S \cup T|$ counts them only once whereas $|S| + |T|$ counts them twice, so we must subtract $|S \cap T|$ to compensate for the double counting.

Returning to our random variables A, B and entropy, we can say that if A, B are independent, i.e. $A \cap B = \emptyset$, then

$$\begin{aligned} H(A,B) &= H(A) + H(B) \\ &= |A| + |B| = |A \cup B|. \end{aligned}$$

In general,

$$H(A,B) = |A \cup B| = |A| + |B| - |A \cap B|.$$

As a consequence,

$$\begin{aligned} H(A,B) &= H(B) + H(A|B) \\ &= |B| + (|A| - |A \cap B|), \end{aligned}$$

which means that

$$H(A|B) = |A| - |A \cap B| = |A \setminus B|,$$

which makes a lot of sense because $H(A|B)$ is the amount of information in A not in B, i.e. not *explained* by B.

Moreover,

$$\begin{aligned} I(A;B) &= H(A) - H(A|B) \\ &= |A| - |A \setminus B| = |A \cap B|, \end{aligned}$$

which, again, makes perfect sense! In fact, $I(A;B)$ can be seen as the (amount of) information *shared* by A and B.

In conclusion:

- $H(A) = |A|$
- $H(A,B) = |A \cup B|$
- $H(A|B) = |A \setminus B|$
- $I(A;B) = |A \cap B|$

See figure 5 for a visual depiction of this interpretation.

From this, one might think that the generalization to the multivariate case is straightforward. For instance: $I(A;B;C) = |A \cap B \cap C|$. Unfortunately, while this is certainly possible, there are problems of interpretation because $I(A;B;C)$ may be negative!

13.1 The Inclusion Exclusion Principle

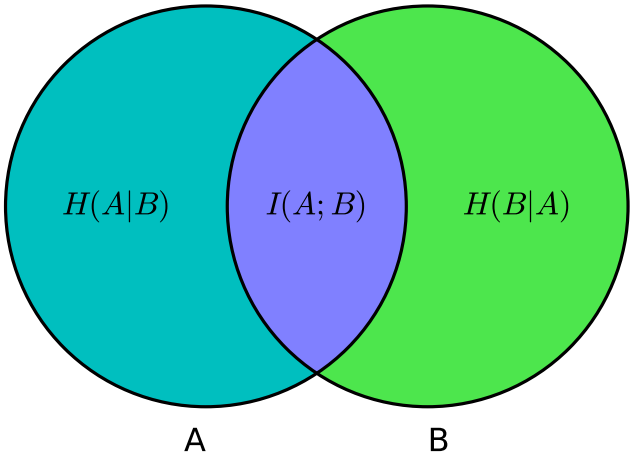
Before talking about the multivariate case, it's better to properly introduce the *Inclusion Exclusion Principle*.

Let A, B be two finite sets and let's indicate the *cardinality* (i.e. the number of elements) of a set X as $|X|$. As we saw, $|A \cup B| = |A| + |B| - |A \cap B|$ because if A and B have elements in common, then $|A| + |B|$ counts those elements twice and subtracting $|A \cap B|$ compensates for that.

The case with 3 variables is similar:

$$\begin{aligned} |A \cup B \cup C| \quad = \quad & |A| + |B| + |C| \\ & -|A \cap B| - |A \cap C| - |B \cap C| \\ & +|A \cap B \cap C| \end{aligned}$$

See picture 6 to see the Inclusion Exclusion Principle for 3 sets in action. The generalization to n sets is easy:



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

Figure 5: *Information Diagram* for visualizing Information Theory relations.

1. We add the cardinalities of all the n sets (i.e. $|X_1|, |X_2|, \ldots, |X_n|$).
2. We subtract the cardinalities of all the possible intersections of 2 sets (i.e. $|X_1 \cap X_2|, |X_1 \cap X_3|, \ldots, |X_2 \cap X_3|, |X_2 \cap X_4|, \ldots\ldots, |X_{n-1} \cap X_n|$).
3. We add the cardinalities of all the possible intersections of 3 sets (i.e. $|X_1 \cap X_2 \cap X_3|, |X_1 \cap X_2 \cap X_4|, \ldots, |X_{n-2} \cap X_{n-1} \cap X_n|$).
4. ... and so on...
5. ... until we add/subtract $|X_1 \cap X_2 \cap \cdots \cap X_n|$.

Written in formula, this becomes

$$\left|\bigcup_{i=1}^{n} X_i\right| = \sum_{i=1}^{n}(-1)^{i+1} \sum_{J:J\subset\{1,\ldots,n\}\wedge|J|=i} \left|\bigcap_{j\in J} X_j\right|$$

where $\wedge$ means "and". We can also lump the two sums together:

$$\left|\bigcup_{i=1}^{n} X_i\right| = \sum_{J:\emptyset\neq J\subset\{1,\ldots,n\}} (-1)^{|J|+1} \left|\bigcap_{j\in J} X_j\right|. \tag{6}$$

This shouldn't be too hard to prove by induction. First of all, note that the sum in expression 6 has $2^n - 1$ terms because the subsets of $\{1, \ldots, n\}$ are 2^n (i.e. the number of all possible binary strings of length n where 1 means *taken* and 0 means *not taken*) and we're excluding $\emptyset$.

Let S_n be the collection of all the non empty subsets of $\{1, \ldots, n\}$. It's easy to see that

$$S_{n+1} = S_n \cup \{K \cup \{n+1\} | K \in S_n\} \cup \{\{n+1\}\}. \tag{7}$$

In fact, $|S_{n+1}| = (2^n - 1) + (2^n - 1) + 1 = 2^{n+1} - 1$, as it should.

Let's try to carry out the inductive step:

$$\begin{aligned}
\left|\bigcup_{i=1}^{n+1} X_i\right| &= \left|\left(\bigcup_{i=1}^{n} X_i\right) \cup \{X_{n+1}\}\right| \\
&= \left|\bigcup_{i=1}^{n} X_i\right| + |X_{n+1}| - \left|\left(\bigcup_{i=1}^{n} X_i\right) \cap X_{n+1}\right| \\
&= \left|\bigcup_{i=1}^{n} X_i\right| + |X_{n+1}| - \left|\bigcup_{i=1}^{n} (X_i \cap X_{n+1})\right|.
\end{aligned}$$

Before proceeding, let's simplify the last term:

$$\begin{aligned}\left|\bigcup_{i=1}^{n}(X_i \cap X_{n+1})\right| &= \sum_{J\in S_n}(-1)^{|J|+1}\left|\bigcap_{j\in J}(X_j \cap X_{n+1})\right| \\ &= \sum_{J\in S_n}(-1)^{|J|+1}\left|\left(\bigcap_{j\in J}X_j\right)\cap X_{n+1}\right| \\ &= -\sum_{J\in\{K\cup\{n+1\}|K\in S_n\}}(-1)^{|J|+1}\left|\bigcap_{j\in J}X_j\right|.\end{aligned}$$

Now, by using equality 7, we can conclude our proof:

$$\begin{aligned}\left|\bigcup_{i=1}^{n+1}X_i\right| &= \left|\bigcup_{i=1}^{n}X_i\right| + |X_{n+1}| - \left|\bigcup_{i=1}^{n}(X_i \cap X_{n+1})\right| \\ &= \sum_{J\in S_n}(-1)^{|J|+1}\left|\bigcap_{j\in J}X_j\right| \\ &\quad + |X_{n+1}| \\ &\quad + \sum_{J\in\{K\cup\{n+1\}|K\in S_n\}}(-1)^{|J|+1}\left|\bigcap_{j\in J}X_j\right| \\ &= \sum_{J\in S_n\cup\{K\cup\{n+1\}|K\in S_n\}\cup\{\{n+1\}\}}(-1)^{|J|+1}\left|\bigcap_{j\in J}X_j\right| \\ &= \sum_{J\in S_{n+1}}(-1)^{|J|+1}\left|\bigcap_{j\in J}X_j\right|.\end{aligned}$$

13.2 Cardinality of Intersections

The Inclusion Exclusion Principle is perfect for evaluating the cardinality of unions, but can we use it for computing $|\bigcap_{i=1}^{n}X_i|$ instead? We could transform unions in intersections (and vice versa) by using the two *De Morgan's Laws*:

$$\left(\bigcup_{i=1}^{n}X_i\right)^C = \bigcap_{i=1}^{n}X_i^C$$
$$\left(\bigcap_{i=1}^{n}X_i\right)^C = \bigcup_{i=1}^{n}X_i^C$$

where X^C denotes the *set complement* of X, i.e. $\Omega\setminus X$, where Ω is the *Universe*, i.e. a set with includes *all* the elements. Basically, X^C is the set of all the elements not in X.

Now we can try to evaluate the cardinality of an intersection:

$$\begin{aligned}
\left|\bigcap_{i=1}^{n} X_i\right| = \left|\left(\bigcup_{i=1}^{n} X_i^C\right)^C\right| &= |\Omega| - \left|\bigcup_{i=1}^{n} X_i^C\right| \\
&= |\Omega| - \sum_{J \in S_n} (-1)^{|J|+1} \left|\bigcap_{j \in J} X_j^C\right| \\
&= |\Omega| - \sum_{J \in S_n} (-1)^{|J|+1} \left|\left(\bigcup_{j \in J} X_j\right)^C\right| \\
&= |\Omega| - \sum_{J \in S_n} (-1)^{|J|+1} \left(|\Omega| - \left|\bigcup_{j \in J} X_j\right|\right) \\
&= |\Omega| - |\Omega| \sum_{J \in S_n} (-1)^{|J|+1} + \sum_{J \in S_n} (-1)^{|J|+1} \left|\bigcup_{j \in J} X_j\right|.
\end{aligned} \tag{8}$$

To complete our derivation, we'll need to simplify the first sum:

$$\begin{aligned}
\sum_{J \in S_n} (-1)^{|J|+1} &= \sum_{J \in S_{n-1} \cup \{K \cup \{n\} | K \in S_{n-1}\} \cup \{\{n\}\}} (-1)^{|J|+1} \\
&= \sum_{J \in S_{n-1}} (-1)^{|J|+1} + \sum_{J \in \{K \cup \{n\} | K \in S_{n-1}\}} (-1)^{|J|+1} + (-1)^{1+1} \\
&= \sum_{J \in S_{n-1}} (-1)^{|J|+1} + \sum_{J \in S_{n-1}} (-1)^{|J \cup \{n\}|+1} + 1 \\
&= \sum_{J \in S_{n-1}} (-1)^{|J|+1} - \sum_{J \in S_{n-1}} (-1)^{|J|+1} + 1 = 1.
\end{aligned}$$

By substituing back into equation 8, we get

$$\left|\bigcap_{i=1}^{n} X_i\right| = \sum_{J \in S_n} (-1)^{|J|+1} \left|\bigcup_{j \in J} X_j\right|. \tag{9}$$

13.3 Multiple Mutual Information

There have been various attempts to generalize the concept of Mutual Information to 3 or more terms.

One generalization is the so-called *Multiple Mutual Information* (*MMI*), *Interaction*, or *Co-information*, defined as follows:

$$I(X_1; \ldots; X_n; X_{n+1}) = I(X_1; \ldots; X_n) - I(X_1; \ldots; X_n | X_{n+1}) \tag{10}$$

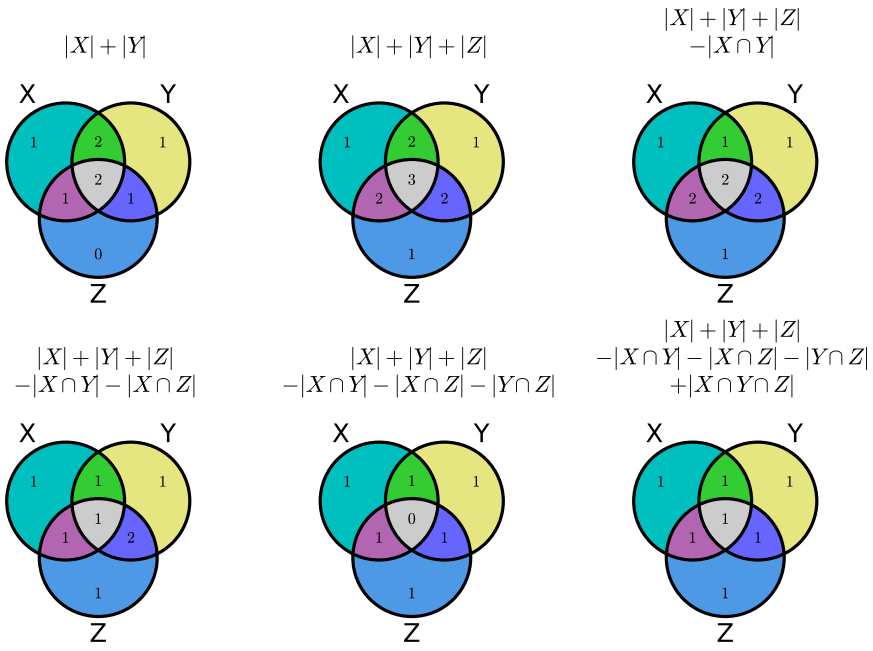


Figure 6: *Inclusion Exclusion Principle* for 3 sets in action. The numbers tell how many times each region is counted. Our goal is to compute $|X \cup Y \cup Z|$ which requires counting each region exactly once.

where

$$I(X_1;\ldots;X_n|X_{n+1}) \quad = \quad \mathbb{E}_{X_{n+1}}[I(X_1;\ldots;X_n|X_{n+1})] \tag{11}$$

There's a subtlety here which we've already come across before. Indeed,

$$\mathbb{E}_{X_{n+1}}[I(X_1;\ldots;X_n|X_{n+1})] = \sum_{x_{n+1}} p(x_{n+1}) I(X_1;\ldots;X_n|X_{n+1}=x_{n+1})$$

so, $I(X_1;\ldots;X_n|X_{n+1})$ inside $\mathbb{E}_{X_{n+1}}[\cdot]$ is actually the random variable $f(X_{n+1})$ where f is the function $x \mapsto I(X_1;\ldots;X_n|X_{n+1}=x)$. The same thing happened before when we wrote

$$H(X|Y) = \mathbb{E}_Y[H(X|Y)] = \sum_y p(y) H(X|Y=y). \tag{12}$$

Also, basically, $I(X_1;\ldots;X_n|X_{n+1}=x)$ can be obtained from $I(X_1;\ldots;X_n)$ by replacing any $p(\cdot)$ and $p(\cdot|\cdot)$ with $p(\cdot|X_{n+1}=x)$ and $p(\cdot|\cdot,X_{n+1}=x)$, respectively. For instance,

$$\begin{aligned}
I(X;Y) &= H(X) - H(X|Y) \\
&= -\sum_x p(x)\log_2 p(x) + \sum_x \sum_y p(x,y)\log_2 p(x|y) \qquad \Longrightarrow \\
I(X;Y|Z=z) &= H(X|Z=z) - H(X|Y,Z=z) \\
&= -\sum_x p(x|Z=z)\log_2 p(x|Z=z) \\
&\quad + \sum_x \sum_y p(x,y|Z=z)\log_2 p(x|y,Z=z).
\end{aligned}$$

Let's go one step further and write the expression for $I(X;Y|Z)$:

$$\begin{aligned}
I(X;Y|Z) &= \sum_z p(Z=z) I(X;Y|Z=z) \\
&= -\sum_z p(Z=z) \sum_x p(x|Z=z)\log_2 p(x|Z=z) \\
&\quad + \sum_z p(Z=z) \sum_x \sum_y p(x,y|Z=z)\log_2 p(x|y,Z=z) \\
&= -\sum_x \sum_z p(x,z)\log_2 p(x|z) + \sum_x \sum_y \sum_z p(x,y,z)\log_2 p(x|y,z) \\
&= H(X|Z) - H(X|Y,Z).
\end{aligned} \tag{13}$$

See picture 7 for a visualization of $I(X;Y|Z)$.

If we interpret $I(X)$ and $I(X|Y)$ as $H(X)$ and $H(X|Y)$, respectively (which makes perfect sense), then definition 10 is also valid for the two-variable case. Note that we defined $I(x)$ as the information contained in $x \in X$, so it makes

sense to define $I(X)$ as $H(X)$. Basically, by convention, something of the form $I(X_1;\ldots;X_k|Y_1,\ldots,Y_h)$ becomes $H(X_1|Y_1,\ldots,Y_h)$ when $k=1$. Note that this makes definition 12 a particular case of definition 11.

According to definition 10, and by our derivation 13 of $I(X;Y|Z)$, the MMI of three variables is

$$\begin{aligned} I(X;Y;Z) &= I(X;Y) - I(X;Y|Z) \\ &= H(X) - [H(X|Y)] - [H(X|Z)] + [H(X|Y,Z)] \\ &= H(X) - [H(X,Y) - H(Y)] - [H(X,Z) - H(Z)] \\ &\quad + [H(X,Y,Z) - H(Y,Z)] \\ &= H(X) + H(Y) + H(Z) \\ &\quad - H(X,Y) - H(X,Z) - H(Y,Z) \\ &\quad + H(X,Y,Z). \end{aligned} \tag{14}$$

Let's rewrite all that in the language of set theory:

$$\begin{aligned} |X \cap Y \cap Z| &= |X \cap Y| - |(X \cap Y) \setminus Z| \\ &= |X| - [|X \setminus Y|] - [|X \setminus Z|] + [|X \setminus (Y \cup Z)|] \\ &= |X| - [|X \cup Y| - |Y|] - [|X \cup Z| - |Z|] \\ &\quad + [|X \cup Y \cup Z| - |Y \cup Z|] \\ &= |X| + |Y| + |Z| - |X \cup Y| - |X \cup Z| - |Y \cup Z| + |X \cup Y \cup Z|. \end{aligned}$$

This is exactly formula 9 for 3 sets!

See picture 7 for a visualization of $I(X;Y;Z)$ and the various terms with 3 random variables. Also, look at picture 8 to see equality 14 "in action".

This is not apparent from our discussion, but the MMI of 3 or more variables may be negative, which makes its interpretation less intuitive. That's probably why MMI is not very popular in Machine Learning.

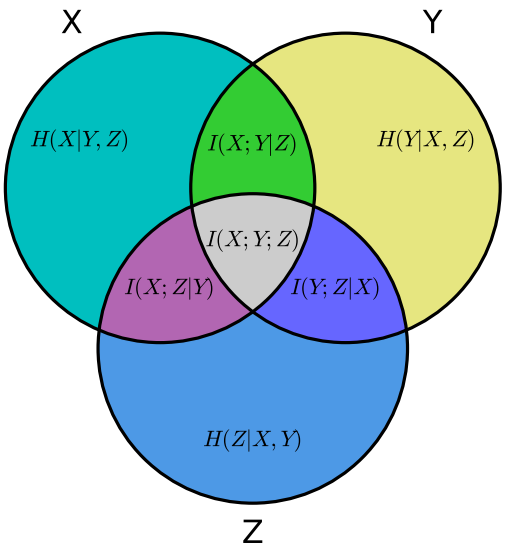
Thanks to equality 9, we can generalize equality 14 as

$$I(X_1;\ldots;X_n) = \sum_{J:\emptyset \neq J \subset \{1,\ldots,n\}} (-1)^{|J|+1} H(X_J),$$

where if $J = \{j_1,\ldots,j_m\}$, then $H(X_J) = H(X_{j_1}, X_{j_2}, \ldots, X_{j_m})$.

The mutual information may also be computed between a set of variables $X_1,\ldots,X_n$ and another variable Y. This is called *Joint Mutual Information* and is defined as

$$\begin{aligned} I(X_1,\ldots,X_n;Y) &= H(X_1,\ldots,X_n) - H(X_1,\ldots,X_n|Y) \\ &= -\sum_{x_1}\cdots\sum_{x_n} p(x_1,\ldots,x_n)\log_2 p(x_1,\ldots,x_n) \\ &\quad + \sum_{x_1}\cdots\sum_{x_n}\sum_{y} p(x_1,\ldots,x_n,y)\log_2 p(x_1,\ldots,x_n|y). \end{aligned}$$



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

Figure 7: *Information Diagram* for 3 random variables.

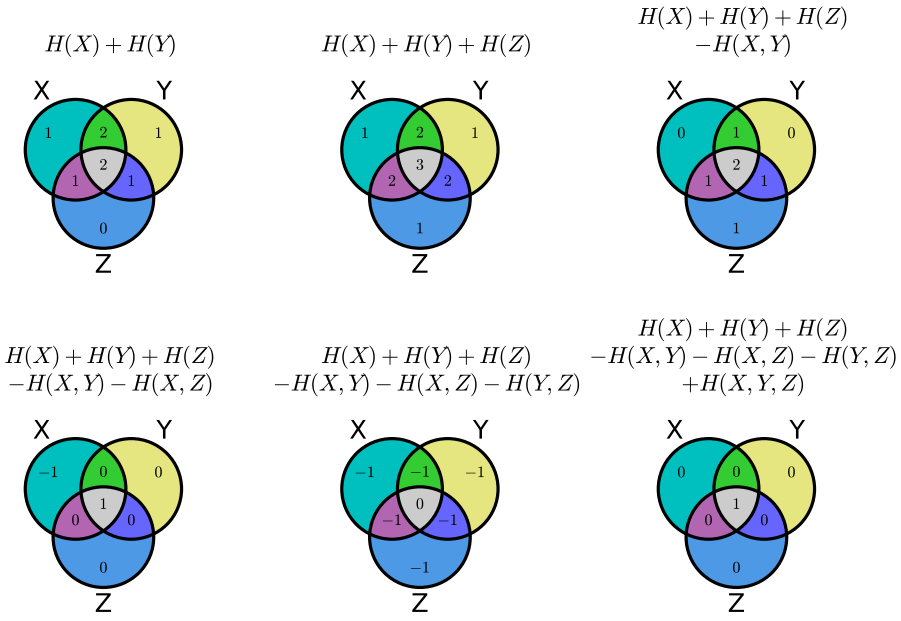


Figure 8: The numbers indicate how many times a subset is counted. By adding $H(X), H(Y), H(Z)$ we overcount some subsets, so we must compensate by subtracting $H(X, Y), H(X, Z), H(Y, Z)$, but now we undercount, so we add $H(X, Y, Z)$ and, finally, we get $I(X; Y; Z)$, i.e. $|X \cap Y \cap Z|$. (Remember that, for any X, Y, Z, $H(X, Y) = |X \cup Y|$ and $H(X, Y, Z) = |X \cup Y \cup Z|$.)

The set theoretic interpretation is

$$|(X_1 \cup \cdots \cup X_n) \cap Y| = |X_1 \cup \cdots \cup X_n| - |(X_1 \cup \cdots \cup X_n) \setminus Y|.$$

In general, any *union* of variables X_i can be represented as a single variable X which has the variables X_i as components: $X = (X_1, \ldots, X_n)$. For instance, with the appropriate definitions,

$$\begin{aligned} H(X_1, \ldots, X_n) &= H(X) \\ H(X_1, \ldots, X_n|Y) &= H(X|Y) \\ I(X_1, \ldots, X_n; Y) &= I(X; Y) \\ I(X; Y_1, \ldots, Y_n) &= I(X; Y) \\ I(X_1, \ldots, X_n; Y_1, \ldots, Y_m) &= I(X; Y) \\ I(X_1, \ldots, X_n; Y_1, \ldots, Y_m; Z_1, \ldots, Z_k) &= I(X; Y; Z). \end{aligned}$$

Basically, we may replace, for instance, "$X_1, \ldots, X_n$" with "X" wherever it appears inside H or I.

Of course, this implies an analogous simplification for formulas with explicit sums over distributions. For instance, here's the Joint Mutual Information again:

$$\begin{aligned} I(X_1, \ldots, X_n; Y) &= I(X; Y) \\ &= H(X) - H(X|Y) \\ &= -\sum_x p(x) \log_2 p(x) + \sum_x \sum_y p(x, y) \log_2 p(x|y). \end{aligned}$$

In this case, we replaced "$X_1, \ldots, X_n$" with "X" and "$x_1, \ldots, x_n$" with "x". If you think you're having a *deja vu*, that's because this is related to proposition 2 and remark 2.

13.4 KL Divergence and Total Correlation

As we saw, the mutual information between X and Y is the KL Divergence between $p(x, y)$ and $p(x)p(y)$. The MMI defined above, though, loses such a simple interpretation. If we generalize mutual information through the KL Divergence we get something simpler:

$$\begin{aligned} I(X_1; \ldots; X_n) &= KL(p(x_1, \ldots, x_n)||p(x_1) \cdots p(x_n)) \\ &= \sum_{x_1} \cdots \sum_{x_n} p(x_1, \ldots, x_n) \log_2 \frac{p(x_1, \ldots, x_n)}{p(x_1) \cdots p(x_n)} \\ &= H(X_1) + \ldots + H(X_n) - H(X_1, \ldots, X_n) \end{aligned}$$

which is called *Total Correlation*. This quantity is always non negative and 0 if and only if the variables are all independent, of course. Note that in this formulation *k-way* "*interactions*", with $1 < k < n$, are completely ignored.

Part II
Applications of Information Theory to Machine Learning

14 Loss function: from KL Divergence to Cross Entropy to Log Likelihood

14.1 Supervised Learning

Let's assume we have some *data* $\mathcal{D} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ where each pair (x_i, y_i) is a sample generated from an *unknown* distribution $p(x, y)$. Note that $\mathcal{D}$ is a *multiset* because it might contain multiple occurrences of the same pair. In *Supervised Learning* we're mainly interested in determining $p(y|x)$ so that we can predict y given x.

To make things more concrete, let's assume that the x_i are images (*raw pixels*) and the y_i are the corresponding *labels* which describe the content of the images.

We can use a *convolutional neural network* (*ConvNet*) with a final *softmax layer*. Let θ be the weights of the ConvNet and f_θ the function computed by the ConvNet with weights θ.

Each pair (x_i, y_i) can be seen as a *Multinoulli* (or *generalized Bernoulli* or *categorical*) distribution d_i over $Y|X = x_i$ which puts all its probability mass on $Y = y_i$, whereas $f_\theta(x_i)$ is the distribution over $Y|X = x_i$ computed by the ConvNet.

The optimal value for θ can be computed as follows:

$$\begin{aligned}
\theta^* &= \operatorname*{argmin}_\theta \sum_{i=1}^{n} KL(d_i || f_\theta(x_i)) \\
&= \operatorname*{argmin}_\theta \sum_{i=1}^{n} [H(d_i, f_\theta(x_i)) - H(d_i)] \\
&= \operatorname*{argmin}_\theta \sum_{i=1}^{n} H(d_i, f_\theta(x_i)) \qquad (15)
\end{aligned}$$

where we dropped the entropy $H(d_i)$ because it doesn't depend on θ. So, we're left with just the cross entropy.

Please note that swapping d_i and $f_\theta(x_i)$ in the cross entropy would be a ***mistake*** because we're optimizing with respect to $f_\theta(x_i)$. In fact, while we saw that $\min_q H(p, q) = H(p, p) = H(p)$ and so we're forcing q to "converge" to p, we can't do the same with $\min_q H(q, p)$, which is, in general, *not* equal to $H(p)$. If this isn't obvious to you, you should reread section 10.

As we already said, d_i puts all the probability mass on $Y = y_i$, so we can

simplify expression 15 as follows:

$$\begin{aligned}\theta^* &= \operatorname*{argmin}_\theta \sum_{i=1}^{n} \left(-\sum_y d_i(y) \log_2 f_\theta(y|x_i) \right) \\ &= \operatorname*{argmin}_\theta \sum_{i=1}^{n} \left(-\log_2 f_\theta(y_i|x_i) \right)\end{aligned}$$

where $f_\theta(y|x_i)$ is *syntactic sugar* (again, programming lingo) for $f_\theta(x_i)(y)$.

It's easy to see that this corresponds to maximizing the *Log Likelihood*:

$$\begin{aligned}\theta^* &= \operatorname*{argmin}_\theta \sum_{i=1}^{n} \left(-\log_2 f_\theta(y_i|x_i) \right) \\ &= \operatorname*{argmax}_\theta \sum_{i=1}^{n} \log f_\theta(y_i|x_i) \\ &= \operatorname*{argmax}_\theta \log \prod_{i=1}^{n} f_\theta(y_i|x_i) \\ &= \operatorname*{argmax}_\theta \log \prod_{i=1}^{n} P(y_i|x_i;\theta) \\ &= \operatorname*{argmax}_\theta \log P(y_1,\ldots,y_n|x_1,\ldots,x_n;\theta) \\ &= \operatorname*{argmax}_\theta \log L(\theta).\end{aligned}$$

14.2 Density Estimation

Now let's consider another case. We have a dataset $\mathcal{D}_2 = \{x_1, \ldots, x_n\}$. Let's assume that the x_i are all samples from a random variable X. We want to fit a *Gaussian* to the data. To do this, we define a family of Gaussians:

$$\mathcal{F} = \left\{ \mathcal{N}\left(x|\mu,\sigma^2\right) \mid \mu \in \mathbb{R}, \sigma^2 \in \mathbb{R}_+ \right\}.$$

Let $\hat{p}$ be the (empirical) distribution represented by $\mathcal{D}_2$ and let's introduce $p_\theta \in \mathcal{F}$ where $\theta = \left(\mu, \sigma^2\right)$. We want to find θ so that p_θ is as close as possible to $\hat{p}$. We can find the optimal parameters by minimizing the KL Divergence

between $\hat{p}$ and p_θ:

$$\begin{aligned}
\theta^* &= \underset{\theta}{\operatorname{argmin}}\, KL(\hat{p}||p_\theta) \\
&= \underset{\theta}{\operatorname{argmin}}[H(\hat{p}, p_\theta) - H(\hat{p})] \\
&= \underset{\theta}{\operatorname{argmin}}\, H(\hat{p}, p_\theta) \\
&= \underset{\theta}{\operatorname{argmin}}\, \mathbb{E}_{X\sim\hat{p}}[I_{p_\theta}(X)] \\
&= \underset{\theta}{\operatorname{argmin}} \sum_x \hat{p}(x)(-\log p_\theta(x)) \\
&= \underset{\theta}{\operatorname{argmax}} \sum_{i=1}^{n} \frac{1}{n} \log p_\theta(x_i) \\
&= \underset{\theta}{\operatorname{argmax}} \sum_{i=1}^{n} \log p_\theta(x_i) \\
&= \underset{\theta}{\operatorname{argmax}} \log \prod_{i=1}^{n} p_\theta(x_i) \\
&= \underset{\theta}{\operatorname{argmax}} \log P(x_1, \ldots, x_n|\theta) \\
&= \underset{\theta}{\operatorname{argmax}} \log L(\theta).
\end{aligned}$$

Once again, the KL Divergence and the cross entropy are equivalent to the maximum log likelihood.

Note that:

- The self-information I contains a log because, as we saw, we want to transform products of probabilities into sums of amounts of information.
- The log is used in the log likelihood because we want to simplify calculations and sums are easier than products.

Different *reasons*, same *result*.

The appealing of cross entropy and log likelihood when using a final softmax layer is that they undo the exp of the softmax and so we get a better performance with *Stochastic Gradient Descent (SGD)* because the risk of *saturation* (i.e. the gradient becoming 0 and the units stopping learning) is reduced.

Note that while maximum log likelihood and cross entropy are equivalent, from a theoretical point of view, the log in the cross entropy has more *justification* because it comes from the additivity of the information rather than being just a convenient trick to simplify calculations.

15 Feature selection

There are two ways to *reduce* the *dimensionality* of a dataset:

feature selection Only a subset of relevant features is retained and the other features are filtered out.

feature extraction The set of original features is transformed into a *reduced* set of features.

Here we're going to talk about *feature selection*.

15.1 Information Gain

Let's assume we have a dataset $\mathcal{D} = \{(x^{(1)}, y^{(1)}), \ldots, (x^{(N)}, y^{(N)})\}$ where $x^{(i)} \in \mathbb{R}^n$ and $y^{(i)} \in \mathbb{R}$. We want to learn the mapping $x \mapsto y$.

The dataset can be seen as generated by $n+1$ random variables: $X_1, \ldots, X_n$, and Y. *Information Gain* represents one of the easiest ways to do *feature selection*. Very simply, we grade each feature X_i according to $I(X_i; Y)$. The more information about the label Y we gain, *on average*, by observing X_i, the more useful X_i is deemed.

Let's define

$$\mathcal{D}\{cond\} = \{(x, y) \in \mathcal{D} | cond((x, y)) \text{ is true}\},$$

i.e. $\mathcal{D}\{cond\}$ is the set of all the pairs for which the condition *cond* (about the random variables $X_1, \ldots, X_n, Y$) is true. In this particular case, the Information Gain is defined as follows:

$$\begin{aligned}
IG(\mathcal{D}, s) &= I(X_s; Y) \\
&= H(Y) - H(Y|X_s) \\
&= -\sum_{y \in Y} p(y) \log_2 p(y) + \sum_{x \in X_s} \sum_{y \in Y} p(x, y) \log_2 p(y|x) \\
&= -\sum_{y \in Y} \frac{|\mathcal{D}\{Y = y\}|}{|\mathcal{D}|} \log_2 \frac{|\mathcal{D}\{Y = y\}|}{|\mathcal{D}|} \\
&\quad + \sum_{x \in X_s} \sum_{y \in Y} \frac{|\mathcal{D}\{Y = y \wedge X_s = x\}|}{|\mathcal{D}|} \log_2 \frac{|\mathcal{D}\{Y = y \wedge X_s = x\}|}{|\mathcal{D}\{X_s = x\}|}.
\end{aligned}$$

The main advantages of Information Gain are simplicity of implementation and low computational cost. The most obvious drawback is that the dependency between features is completely ignored, so some of the features selected because of their high score may carry *redundant* information. For instance, if X_1 has one of the highest scores among the variables and $X_2 = X_1$, then, probably, both X_1 and X_2 will be selected even though they contain the exact same information about Y.

15.2 Joint Mutual Information

A better way to choose the best k features consists in maximizing the Joint Mutual Information. In theory, we'd like to grade a subset of features $F =$

$(F_1, \ldots, F_k)$ by

$$I(F_1, \ldots, F_k; Y) = H(F_1, \ldots, F_k) - H(F_1, \ldots, F_k|Y)$$

or, equivalently,

$$I(F; Y) = H(F) - H(F|Y).$$

Unfortunately, this requires the estimation of a high-dimensional probability density function, which is very expensive and requires lots of data.

Some methods try to approximate the Joint Mutual Information, while others try to approximate even more general measures[1].

16 EM Algorithm

16.1 Bonus

The *EM Algorithm* seems completely unrelated to Information Theory, but there's an interesting connection that will be revealed at the end. I don't know whether this connection is strong enough to warrant two sections about the EM Algorithm. In case it isn't, take this part as a *bonus*!

16.2 Definition

Let's assume we want to fit a model to some data by maximizing the log likelihood. We have a sample $x \in X$, which represents our dataset, and we want to find

$$\theta^* = \operatorname*{argmax}_\theta L(\theta; x) = \operatorname*{argmax}_\theta \log p(x|\theta).$$

Sometimes, the model can be simplified by introducing a *latent* (i.e. unobservable) random variable Z:

$$L(\theta; X) = \log p(x|\theta) = \log \sum_z p(x, z|\theta). \tag{16}$$

We may try to find a *closed-form* formula for θ^* or, if not possible or convenient, we could use an iterative optimization algorithm like (*Stochastic*) *Gradient Descent*. The problem is that $L(\theta; X)$ and $\nabla_\theta L(\theta; X)$ may be very difficult to compute. Often, $p(x, z|\theta)$ factorizes so the log, if only it was inside the sum, would transform multiplications into additions simplifying computations considerably.

The *EM Algorithm* solves the problem by working with the *complete data log likelihood*:

$$L(\theta; X, Z) = \log p(x, z|\theta).$$

This is similar to bringing the log inside the sum in equation 16 which, as we said, may simplify computations. Since we don't really have a sample $z \in Z$, being Z unobservable, $L(\theta; X, Z)$ is really a function of Z and θ. The EM Algorithm

gets rid of Z by computing the expectation of $L(\theta; X, Z)$ with respect to Z and then maximizing the resulting function with respect to θ.

We set θ^1 to some initial value and then we repeat two steps, which give the name to the algorithm, until convergence:

1. initialize θ^1
2. **for** $t = 1, 2, \ldots,$ until *convergence*:
   (a) **Expectation (E) Step:**
   $Q(\theta|\theta^t) = \mathbb{E}_{Z|X,\theta^t}[L(\theta; X, Z)] = \sum_z p(z|x, \theta^t) \log p(x, z|\theta)$
   (b) **Maximization (M) Step:**
   $\theta^{t+1} = \text{argmax}_\theta \, Q(\theta|\theta^t)$

16.3 Example

Let's try to fit a *mixture model* to a dataset $\mathcal{D} = \{x_1, \ldots, x_n\}$ generated from n *i.i.d.* (independent and identically distributed) random variables. In other words, $X = (X_1, \ldots, X_n)$ where $X_i \sim X_j$ for all i, j and $p(x) = p(x_1) \cdots p(x_n)$. Thus,

$$\begin{aligned} L(\theta; X) &= \log p(x|\theta) \\ &= \log \prod_{i=1}^{n} p(x_i|\theta) \\ &= \sum_{i=1}^{n} \log p(x_i|\theta) \end{aligned}$$

and, if $Z = (Z_1, \ldots, Z_n)$,

$$\begin{aligned} L(\theta; X, Z) &= \log p(x, z|\theta) \\ &= \log p(x_1, \ldots, x_n, z_1, \ldots, z_n|\theta) \\ &= \log \prod_{i=1}^{n} p(x_i, z_i|\theta) \\ &= \sum_{i=1}^{n} \log p(x_i, z_i|\theta). \end{aligned}$$

IMPORTANT: From now on, until otherwise indicated, x will refer to *individual* samples and *not* to the entire dataset.

A mixture model is an *average* of models:

$$p(x|\theta) = \sum_z p(x, z|\theta) = \sum_z p(z)p(x|z, \theta) = \mathbb{E}_Z[p(x|Z, \theta)].$$

Let's assume we're averaging K models. Then Z must take K different values, which means that $Z \sim Multinoulli(\pi_1, \ldots, \pi_K)$. We may assume that

the K models have K different parameters $\phi_1, \ldots, \phi_K$. In other words, for $i = 1, \ldots, K$, the i-th model has density $p(x|\phi_i)$.

Let's define $\theta = (\phi, \pi)$ and compute $Q(\theta|\theta^t)$:

$$\begin{aligned}
Q(\theta|\theta^t) &= \mathbb{E}_{Z|X,\theta^t}[L(\theta; X, Z)] \\
&= \mathbb{E}_{Z|X,\theta^t}\left[\log \prod_{i=1}^{n} p(x_i, Z_i|\theta)\right] \\
&= \mathbb{E}_{Z|X,\theta^t}\left[\sum_{i=1}^{n} \log p(x_i, Z_i|\theta)\right] \\
&= \sum_{i=1}^{n} \mathbb{E}_{Z|X,\theta^t}\left[\log[p(Z_i|\pi)p(x_i|Z_i, \phi)]\right] \\
&= \sum_{i=1}^{n} \mathbb{E}_{Z_i|X_i,\theta^t}\left[\log[p(Z_i|\pi)p(x_i|Z_i, \phi)]\right] \\
&= \sum_{i=1}^{n}\sum_{j=1}^{K} p(Z_i = j|x_i, \theta^t) \log[p(Z_i = j|\pi)p(x_i|Z_i = j, \phi)] \\
&= \sum_{i=1}^{n}\sum_{j=1}^{K} p(Z_i = j|x_i, \theta^t) \log[\pi_j p(x_i|\phi_j)].
\end{aligned}$$

Each term $r_{ij} = p(Z_i = j|x_i, \theta^t)$ is called *responsibility* because it measures how much each of the K models is responsible for (generating) x_i. We can compute r_{ij} by using *Bayes' Rule*:

$$\begin{aligned}
r_{ij} &= p(Z_i = j|x_i, \theta^t) \\
&= \frac{p(Z_i = j, x_i|\theta^t)}{\sum_{j=1}^{K} p(Z_i = j, x_i|\theta^t)} \\
&= \frac{p(x_i|Z_i = j, \phi^t)p(Z_i = j|\pi^t)}{\sum_{j=1}^{K} p(x_i|Z_i = j, \phi^t)p(Z_i = j|\pi^t)} \\
&= \frac{p(x_i|\phi_j^t)\pi_j^t}{\sum_{j=1}^{K} p(x_i|\phi_j^t)\pi_j^t}.
\end{aligned}$$

Basically, the algorithm is as follows:

1. initialize ϕ, π
2. **until** convergence:
   (a) **for** $(i, j) \in \{1, \ldots n\} \times \{1, \ldots, K\}$:
   $$r_{ij} \leftarrow \frac{p(x_i|\phi_j)\pi_j}{\sum_{j=1}^{K} p(x_i|\phi_j)\pi_j}$$
   (b) $(\phi, \pi) \leftarrow \text{argmax}_{\phi,\pi}\left[\sum_{i=1}^{n}\sum_{j=1}^{K} r_{ij} \log[\pi_j p(x_i|\phi_j)]\right]$

As we said before, depending on the form of the K models, we may be able to find closed-form formulas for the optimal values of the parameters. If that's not possible, we can use an iterative optimization method such as (Stochastic) Gradient Descent. Either way, we'll need to compute or approximate $\nabla_\theta Q(\theta|\theta^t)$.

17 EM Algorithm: why does it work?

The EM Algorithm maximizes $\mathbb{E}_{Z|X,\theta^t}[L(\theta; X, Z)]$, but we should be maximizing $L(\theta; X)$ instead. The reason the EM Algorithm works is that $\mathbb{E}_{Z|X,\theta^t}[L(\theta; X, Z)]$ is a lower bound of $L(\theta; X)$ so maximizing it has the effect of maximizing $L(\theta; X)$ as well.

The EM Algorithm is a particular instance of a method called *MM Algorithm*, which stands for *Majorization Minimization* in case of minimization and *Minorization Maximization* in case of maximization.

17.1 The MM Algorithm

As we said, the MM Algorithm may be used for both minimization and maximization but since we're interested in maximizing $L(\theta; X)$ here, we'll focus on the maximization version which consists of two steps:

1. Minorization
2. Maximization

Let $f(\theta)$ be the function we want to maximize with respect to θ. We say that a function $Q(\theta|\theta_0)$ *minorizes* $f(\theta)$ if

$$\forall \theta \in \text{Dom(f)},\ Q(\theta|\theta_0) \leq f(\theta)$$
$$Q(\theta_0|\theta_0) = f(\theta_0).$$

In words, Q minorizes f if Q is a lower bound of f and touches f in one point.

For an example, go back to section 6 and observe that we used minorization to prove Jensen's Inequality. In particular, see picture 4.

The MM Algorithm is simple:

1. initialize θ
2. **until** convergence:
   (a) **Minorization Step:**
   "Compute" $Q(t|\theta)$ such that $Q \leq f$ and $Q(\theta|\theta) = f(\theta)$
   (b) **Maximization Step:**
   $\theta \leftarrow \text{argmax}_t\, Q(t|\theta)$

The maximization step never decreases f. Indeed, if $\theta^* = \text{argmax}_t\, Q(t|\theta)$, then

$$f(\theta^*) \geq Q(\theta^*|\theta) \geq Q(\theta|\theta) = f(\theta)$$

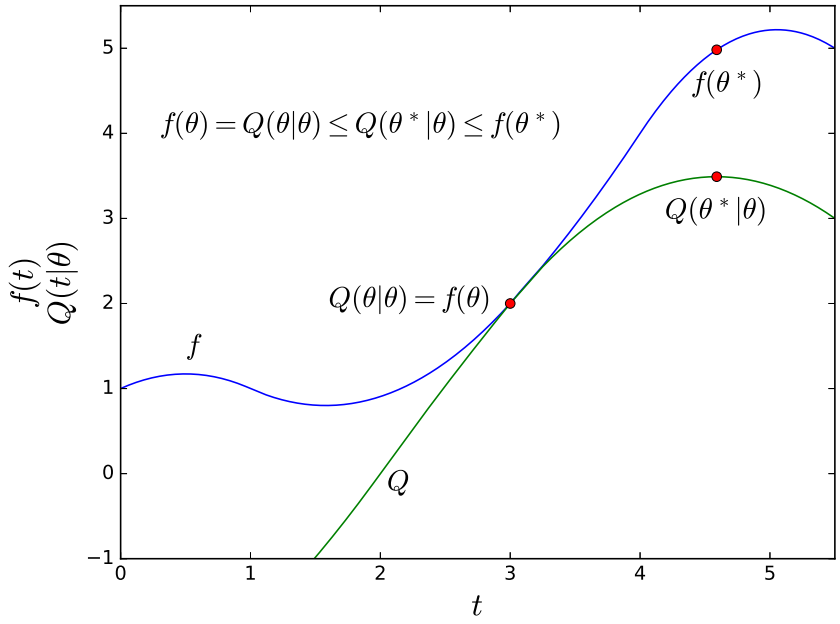


Figure 9: Intuition behind the MM Algorithm.

Moreover, if $Q(\theta^*|\theta) > Q(\theta|\theta)$, then we have an actual increase in f. See picture 9.

Let's make two important observations:

1. Each Minorization Step may compute different functions Q: they don't have to have the same form.

2. The MM Algorithm works even if we don't maximize $Q(t|\theta)$ in the Maximization Step, but just increase it.

The second observation is important because, usually, we're only able to find *local* maxima.

17.2 EM is an instance of MM

By proving that the EM Algorithm is an instance of the MM Algorithm, we prove that the EM Algorithm works.

The E step of the EM Algorithm computes

$$Q(\theta|\theta^t) = \mathbb{E}_{Z|X,\theta^t}[L(\theta; X, Z)] = \sum_z p(z|x, \theta^t) \log p(x, z|\theta)$$

and the M step maximizes it with respect to θ. We can prove that $Q(\theta|\theta^t)$ is a lower bound of $L(\theta; X)$, but they don't touch as required by the MM Algorithm.

Nonetheless, we can prove that optimizing $Q(\theta|\theta^t)$ is equivalent to optimizing a lower bound $Q(q, \theta|\theta^t)$ that *does* touch $L(\theta; X)$ at θ^t.

Since log is concave (i.e. $-\log$ is convex), we can use Jensen's inequality (see section 6) to push the log inside the sum in $L(\theta; X)$ by introducing a distribution $q(z|\theta^t)$ which depends on the parameters at time t:

$$\begin{aligned}
L(\theta; X) &= \log p(x|\theta) \\
&= \log \sum_z p(x, z|\theta) \\
&= \log \sum_z q(z|\theta^t) \frac{p(x, z|\theta)}{q(z|\theta^t)} \\
&= \log \mathbb{E}_{Z\sim q}\left[\frac{p(x, Z|\theta)}{q(Z|\theta^t)}\right] \\
&\geq \mathbb{E}_{Z\sim q}\left[\log \frac{p(x, Z|\theta)}{q(Z|\theta^t)}\right] = Q(q, \theta|\theta^t).
\end{aligned}$$

This derivation is valid for any positive q (it must be positive because it appears in the denominator). We saw that Jensen's Inequality becomes an equality when the random variable is constant, i.e. when

$$\frac{p(x, z|\theta)}{q(z|\theta^t)} = c,$$

where c is a constant. Let's assume that $\theta = \theta^t$ and solve for $q(z|\theta)$:

$$\begin{aligned}
p(x, z|\theta) &= cq(z|\theta) \\
\sum_z p(x, z|\theta) &= c \sum_z q(z|\theta) \\
p(x|\theta) &= c
\end{aligned}$$

thus

$$q(z|\theta) = \frac{p(x, z|\theta)}{c} = \frac{p(x, z|\theta)}{p(x|\theta)} = p(z|x, \theta).$$

It's easy to verify that $Q(q, \theta^t|\theta^t) = L(\theta^t; X)$ for $q = p(z|x, \theta^t)$:

$$\begin{aligned}
Q(p(z|x, \theta^t), \theta^t|\theta^t) &= \mathbb{E}_{Z\sim p(z|x,\theta^t)}\left[\log \frac{p(x, Z|\theta^t)}{p(Z|x, \theta^t)}\right] \\
&= \mathbb{E}_{Z\sim p(z|x,\theta^t)}\left[\log p(x|\theta^t)\right] \\
&= \log p(x|\theta^t) = L(\theta^t; X).
\end{aligned}$$

This means that we can optimize $L(\theta; X)$ by doing *coordinate ascent* (i.e. optimizing with respect to one coordinate at a time) on $Q(q, \theta)$ (we dropped "$|\theta^t$" for convenience):

1. initialize θ

2. **until** convergence:

   (a) **E Step / Minorization:**
   $q \leftarrow \text{argmax}_q \, Q(q, \theta)$ [equivalent to: $q \leftarrow p(z|x, \theta)$]

   (b) **M Step / Maximization:**
   $\theta \leftarrow \text{argmax}_\theta \, Q(q, \theta)$

As we can see, the E Step of the EM Algorithm is equivalent to optimizing with respect to q. In fact, in the EM Algorithm we compute $r_c = p(Z = c|x, \theta^t)$, where the vector r is the optimum value for q under the current parameter θ^t. This is also equivalent to the Minorization step in the MM Algorithm. In fact, now $Q(r, \theta)$ touches $L(\theta; X)$ at the point θ (which we also called θ^t).

The problem is that the EM Algorithm doesn't seem to use $Q(q, \theta|\theta^t)$. Indeed (see 2a),

$$Q(\theta|\theta^t) = \mathbb{E}_{Z|X,\theta^t}[L(\theta; X, Z)]$$

is different from

$$\begin{aligned} Q(r, \theta|\theta^t) &= \mathbb{E}_{Z|X,\theta^t}\left[\log \frac{p(x, Z|\theta)}{p(Z|x, \theta^t)}\right] \\ &= \mathbb{E}_{Z|X,\theta^t}[L(\theta; X, Z)] + H(r). \end{aligned}$$

By noticing that $H(r)$ doesn't depend on θ, we can conclude that

$$\theta^* = \underset{\theta}{\text{argmax}}\, Q(\theta|\theta^t) = \underset{\theta}{\text{argmax}}\, Q(r, \theta|\theta^t)$$

so it's as if the EM Algorithm was using $Q(r, \theta|\theta^t)$.

17.3 Information Theory interpretation

We proved that $Q(q, \theta|\theta^t) \leq L(\theta; X)$ and $Q(q, \theta^t|\theta^t) = L(\theta^t; X)$ when $q = p(z|x, \theta^t)$. This last equality has a nice interpretation:

$$\begin{aligned} L(\theta^t; X) - Q(q, \theta^t|\theta^t) &= \log p(x|\theta^t) - \mathbb{E}_{Z\sim q(z|\theta^t)}\left[\log \frac{p(x, Z|\theta^t)}{q(Z|\theta^t)}\right] \\ &= \log p(x|\theta^t) - \mathbb{E}_{Z\sim q(z|\theta^t)}\left[\log \frac{p(Z|x, \theta^t)p(x|\theta^t)}{q(Z|\theta^t)}\right] \\ &= \log p(x|\theta^t) - \mathbb{E}_{Z\sim q(z|\theta^t)}\left[\log \frac{p(Z|x, \theta^t)}{q(Z|\theta^t)}\right] - \log p(x|\theta^t) \\ &= \mathbb{E}_{Z\sim q(z|\theta^t)}\left[\log \frac{q(Z|\theta^t)}{p(Z|x, \theta^t)}\right] = KL(q(z|\theta^t)||p(z|x, \theta^t)), \end{aligned}$$

so maximizing the lower bound is equivalent to minimizing the KL Divergence between $q(z|\theta^t)$ and $p(z|x, \theta^t)$. In the E step of the EM Algorithm, we find the optimum $q = p(z|x, \theta^t)$, but in some cases computing $p(z|x, \theta^t)$ is intractable. A practical solution consists in constraining q to be of a particular tractable

form (e.g. factorized) and then minimizing the KL Divergence or, equivalently, maximizing $Q(q, \theta)$ with respect to q.

The M step can be generalized as well by just increasing $Q(q, \theta)$ with respect to θ without necessarily finding the optimum.

References

[1] Jorge R. Vergara and Pablo A. Estévez. A review of feature selection methods based on mutual information. *CoRR*, abs/1509.07577, 2015.