9 Information Theory

This chapter provides a brief introduction to some of the fundamental concepts in Information Theory, as it relates to machine learning.

> *Aside:*
> The field of Information Theory was originally developed to model the capacity of a noisy channel through which signals are coded and transmitted, but its applications extend far beyond those origins. The landmark paper in the field, called *A Mathematical Theory of Communication*, by Claude Shannon, was published in 1948; it is very accessible and highly recommended.

9.1 Entropy

At the heart of information theory is the notion of *entropy*. For our purposes, entropy provides a measure of uncertainty associated with a radnom variable or random process. For a discrete random variable $x \in \{1, ..., K\}$, with probabilities $P_i \equiv P(x = i)$, entropy (often denoted H) is defined as

$$\begin{aligned} H &= \sum_{i=1}^{K} P_i \log_2 \frac{1}{P_i} \\ &= -\sum_{i=1}^{K} P_i \log_2 P_i \end{aligned} \tag{1}$$

We sometimes write $H(x)$ to explicitly indicate the random variable in question. Here, with the use of the logarithm in base 2, H is measured in bits. Often in machine learning you will see the natural logarithm uses instead (i.e., base e) in which case H is in nats (and hence a constant multiple of H in bits). You may recognize this notion of entropy from other fields of study, in particular, from statistical thermodynamics.

The idea behind entropy is that rare events are more surprising – they carry more information. Shannon suggested a quantitative measure of information (surprise), namely, $\log_2 \frac{1}{P_i}$. Intuitively, an event with probability 1 (i.e., which occurs without fail) is not very surprising and hence carries little information. An unexpected event, one with low probability will have a larger value of $\log_2 \frac{1}{P_i}$, i.e., more information. Equation (1) can be seen as the expected information associated with an observation of the outcome of an event, i.e., $\mathbb{E}_x[-\log_2 P(x)]$. One important property of entropy is that when random variables x and y are independent, $P(x, y) = P(x)\, P(y)$, then it is easy to show that the entropy of the joint distribution is simply the sum of the entropies of x and y, as one would hope.

It's useful to consider a concrete example. Suppose you flip a fair coin with two equally likely outcomes (heads and tails). In that case $P_1 = P_2 = \frac{1}{2}$, and so the entropy becomes $H = -2\,\frac{1}{2}\log_2 \frac{1}{2} = 1$. So communicating the outcome of a fair coin toss conveys 1 bit of information. If, on the other hand, the coin always lands heads side up, then $P_1 = 1$ and H is easily shown to be 0; i.e., the outcome conveys no information since the outcome was already known with certainty. If

one has a random variable with 8 possible outcomes, all of which are equally likely (i.e., $P_i = \frac{1}{8}$), then its entropy is 3 bits, but if some outcomes are more likely that others, there is less uncertainty about the outcome, and the entropy is lower (but bounded below by 0). As a practical matter, note that the limit of $P_i \log P_i$ is 0 as $P_i \to 0$, so when computing the sum in (1) one can ignore terms for which $P_i = 0$.

In the original work, the aim was to model a data source as a random process, and characterize the amount of information, and hence the necessary code length, in terms of the probability distribution over the symbols in the message. Importantly, Shannon proved the seminal theorem that established entropy as a lower bound on the expected number of bits needed to code data from a source for which the symbols are generated (independently) with probabilities P_i. This further showed that the code length for each should depend on its probability of occurrence, i.e., $l_i = log_2 P_i$ for the Ith symbol (as in Huffman codes).

9.2 Conditional and Relative Entropy

Conditional entropy is the expected entropy in one random variable x, when conditioned on a random variable y:

$$\begin{aligned} H(x|y) &= -\sum_{i,j} P(x_i, y_j) \log_2 P(x_i|y_j) \\ &= -\sum_j P(y_j) \sum_i P(x_i|y_j) \log_2 P(x_i|y_j) \\ &= \sum_j P(y_j)\, H(x|y_j) \end{aligned} \tag{2}$$

This is the expectation, under the distribution $P(y)$, of the entropy of x given the state of y. If x is completely determined by y, i.e., If everything about x is known with certainty once we know the value of y, then the conditional entropy is zero. One can also show that x and y are independent if and only if the conditional entropy satisfies $H(x|y) = H(x)$.

Relative entropy, also called the Kullback-Leibler (KL) divergence is a measure of the difference between two distributions. It plays an essential role in many areas of machine learning, especially in the formulation of variational methods. Given two probability distributions, $Q(x)$ and $P(x)$, the KL divergence between Q and P is

$$\begin{aligned} D_{KL}(Q(x) \,||\, P(x)) &= \sum_i Q(x_i) \log_2 \frac{Q(x_i)}{P(x_i)} \\ &= \sum_i Q(x_i) \log_2 Q(x_i) \;-\; \sum_i Q(x_i) \log_2 P(x_i) \\ &= -H(Q) - \mathbb{E}_Q[\log_2 P] \,. \end{aligned} \tag{3}$$

One can show that the KL divergence is non-negative. It is zero if and only if the two distributions behave identically, i.e., $Q(x) = P(x)$. Also note that it is asymmetric measure, $D_{KL}(Q \,||\, P) \neq D_{KL}(P \,||\, Q)$, so it is not a similarity metric.

In terms of coding, we wanted to transmit x to a receiver, but instead of P, we used Q to derive the coding scheme (i.e., a code that is optimal under Q, rather than P). Then the relative entropy is a measure of (lower bound on) the additional information (bits) that one must transmit compared to an optimal code based on P.

9.3 Mutual Information

Mutual information is one of the most fundamental concepts in information theory. It is a measure of the information shared by two random variables; i.e., a measure of how much about the state of one such variable is known when conditioned on the state of the other. It is defined in terms of entropy and conditional entropy, i.e.,

$$I(x;y) \;=\; H(x) - H(x|y) \;=\; H(y) - H(y|x)\;. \tag{4}$$

So, if knowing the value of y tells us everything about the value of x, then the mutual information is equal to the entropy of x. If the two variables are independent then the mutual information is easily shown to be 0.

You might also recognize mutual information as the same thing as information gain, which we used for learning split functions in decision trees. Remember that, when we evaluated hypothetical split tests, the goal was to a split that reduced uncertainty. In the context of decision trees it was defined as

$$IG(\mathcal{D}_j, t_j) \;=\; H(\mathcal{D}_j) - \frac{N_L}{N_j} H(\mathcal{D}_L) - \frac{N_R}{N_j} H(\mathcal{D}_R)\;, \tag{5}$$

where the data $\mathcal{D}_j$ at node j, with N_j points, is partitioned by the split function t_j into left and right sets, $\mathcal{D}_L$ and $\mathcal{D}_R$, with N_L and N_R points respectively. This is simply the entropy over the class label for data at node j, minus the conditional entropy over the class labels given the split. To see this, note that $\frac{N_L}{N_j}$ and $\frac{N_R}{N_j}$ are simply the probabilities of taking the left and right branches, respectively. $H(\mathcal{D}_L)$ is the entropy over the class labels conditioned on taking the left branch; and similarly for $H(\mathcal{D}_R)$. The sum of these two branch entropies, weighted by the probability of taking one branch of the other, is the conditional entropy for the class labels given the split (with some abuse of notation):

$$IG(\mathcal{D}_j, t_j) \;=\; H(\mathcal{D}_j) - H(\mathcal{D}_j \,|\, t_j)\;. \tag{6}$$

Note that if the split function perfectly separated training samples (so all samples in each of the children had the same label), then one would expect the left and right branch entropies to be zero, thereby maximizing information gain.

There is also a link between mutual information and the KL divergence. One can show that the mutual information between x and y, with marginal and joint probability distributions, $P(x)$, $P(y)$ and $P(x,y)$, can be expressed in terms of the KL divergence as follows:

$$I(x;y) \;=\; D_{KL}(P(x,y) \,||\, P(x)\,P(y))\;. \tag{7}$$

It is easy to see here that the mutual information is therefore zero if x and y are independent. In general $I(x; y) \geq 0$. It can also be shown that mutual information satisfies

$$I(x; y) \;=\; \mathbb{E}_y[D_{KL}(P(x|y) \,||\, P(x))] \;. \tag{8}$$

Finally, we note that mutual information plays a particularly important role when working with continuous variables (although we will not use it for this purpose in this course). With continuous random variables it is natural to extend the definition of entropy above to the integral expressing the expected value of the logarithm (base 2) of the density function. But there are issues due to the fact that a continuous random variable can take on an infinite number of possible states, and hence the entropy does not satisfy the same properties as it does in the discrete case. Mutual information allows one to specify how much information is available about a real-valued random variable conditioned on the existence of noise in the channel through which it is communicated.

9.4 Cross Entropy

The *cross entropy* between two distributions Q and P, is given by

$$H \;=\; -\sum_i Q_i \log_2 P_i \;. \tag{9}$$

It is the expected 'surprise' of a random variable distributed according to P with expectation with respect to Q. You might note that this quantity shows up in the definition of the KL divergence above (3).

Cross entropy is particularly important in machine learning in the guise of the cross entropy loss (also known as the log loss) in the formulation of classification problems. We saw it above when learning the weights for logistic regression. It is also widely used for training deep neural nets. For this problem, we can view Q as the probability distribution over class labels for a training sample, and P as the probability distribution over labels predicted by the learned model. In the case of a binary classification problem, the cross-entropy loss for logistic regression in the last chapter took the form:

$$L(\mathbf{w}) \;=\; -\sum_{i=1}^{N} y_i \, \log P(c_1|\mathbf{x}_i) \;+\; (1 - y_i) \, \log(1 - P(c_1|\mathbf{x}_i)) \;. \tag{10}$$

where of course P is the distribution over the two classes provided by the logistic regressor, and the distribution Q places all the probability mass on whichever the training label is, y_i for the ith sample.

