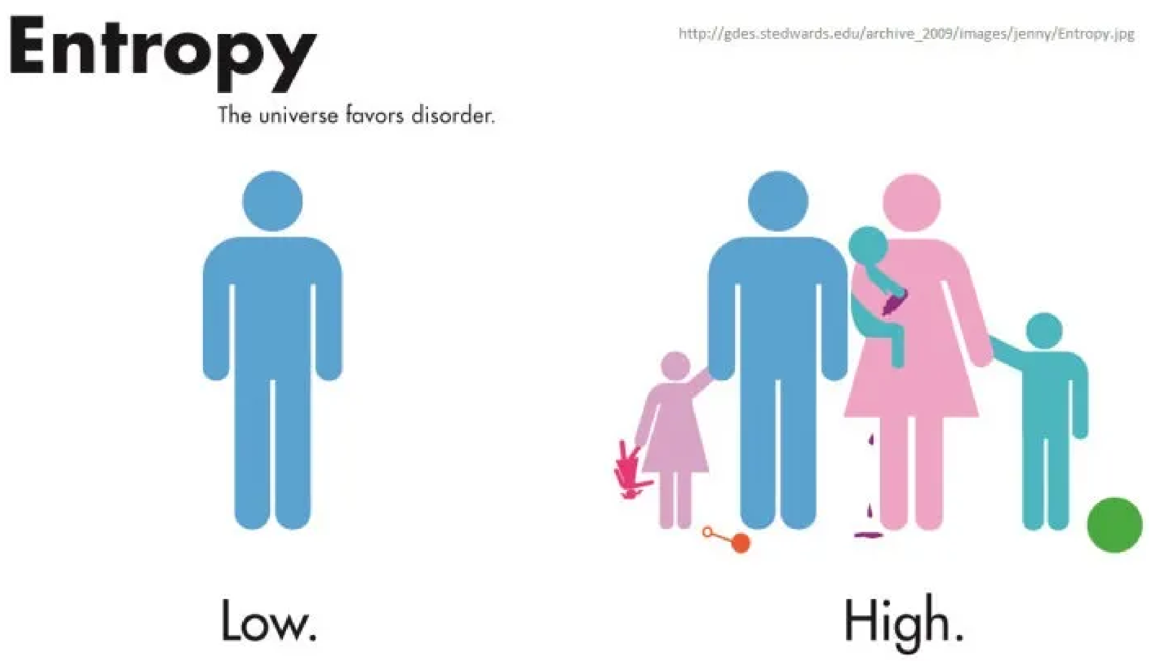
Day 5 — Entropy, Relative Entropy, and Cross Entropy

Tzu-Chi Lin · Follow

Published in 30 days of Machine Learning

5 min read · Dec 21, 2018


Entropy
The universe favors disorder.
http://gdes.stedwards.edu/archive_2009/images/jenny/Entropy.jpg
Low.
High.

The universe favors disorder.

Today we'll focus on the theory of entropy. Understand the intuition of entropy, and how it relates to logistic regression. We'll cover from entropy, KL divergence, to cross entropy.

Entropy is introduced in thermodynamic system from physics. It then be used in many fields, including statistical mechanics, biology, and information theory. In machine learning, we use entropy from information theory. So, what is entropy? And what is it to do with machine learning?

Entropy

First of all, consider a random variable x and we want to know how much information is gained when we observe a specific value for this variable. This amount of information can be viewed as **degree of surprise** on learning the value of x. Denote I(x) as the information content. Suppose x and y are independent and identically distributed (iid), then the information gain from observing both of them should be the sum of the information gained from each of them separately, which means I(x,y) = I(x) + I(y). Denote p(x) as the probability distribution of x. We know that p(x,y) = p(x)p(y) since they are iid. From these two relationships, we can deduct that h(x) is the logarithm of p(x) and we have

$$I(x) = -logp(x)$$

Information Content

where the negative sign ensures that information is non-negative.

For a random variable X, the expectation of information content E[I(X)] is called entropy. Denote H(X) as the entropy of X, we have

$$H(X) = E[I(X)] = \sum_{x \in X} I(x)p(x) = -\sum_{x \in X} p(x)logp(x)$$

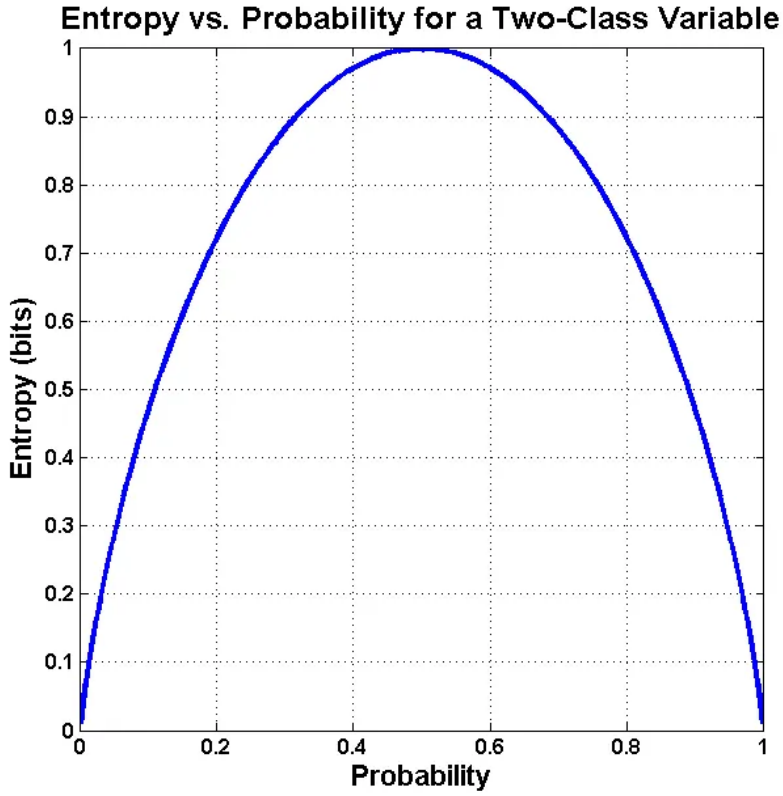
Formula of Entropy

Note that

$$lim_{p \to 0} plnp = 0$$

so we take p(x)lnp(x)=0 whenever we have a value for x such that p(x)=0.

More intuitive, we can think entropy as the **degree of disorder**. When the probability of x is 1 or 0, it is most order, in other words, most consistent, hence it's entropy is 0. On the other hand, when the probability of x is 0.5, it is most disorder (inconsistent). Hence, its entropy is 1.



Entropy(bits) vs Probability

Here we use 2 as our logarithm base, which is called unit of 'bits'. Since entropy deals with bit encoding in the first place in information theory (noiseless coding theorem), it's natural to use 2 for bits as logarithm base. We can also use natural logarithms in defining entropy. In this case, the entropy is measured in units of 'nats' instead of bits.

We use a simple example to show entropy. Consider two students, student A always fails the test, and student B always pass the test. Denote probability p as the probability of passing the test. If they both pass the test this time, then we have

$$P(A) = 0.1 \rightarrow I(A) = -log_2(0.1) = 3.32$$
$$P(B) = 0.9 \rightarrow I(B) = -log_2(0.9) = 0.15$$

Not surprisingly, student A has more information for this test, since A passes the test this time and he always failed the test before. On the other hand, student B always passes the test, it is common for him to pass the test, there's no much information for this test when he passes the test again.

Let's take a look at entropy now.

$$H_A(X) = -[P(A)log(P(A)) + (1 - P(A))log(1 - P(A))] = 0.47$$
$$H_B(X) = -[P(B)log(P(B)) + (1 - P(B))log(1 - P(B))] = 0.47$$

They have same entropy. Since they have the same **degree of disorder** in this case. For student A, it has 10% to pass and 90% to fail. For student B, it has 90% to pass and 10% to fail. It is symmetric in this case. We can also tell from the graph above that entropy is symmetrical.

Relative Entropy (KL Divergence)

Relative entropy, also called KL divergence (Kallback-Leiber divergence), is **the measurement of the distance of two probability distributions**, where p is the true distribution and q is the approximating distribution we have modelled. Define KL divergence as

$$\begin{aligned}D_{KL}(p||q) &= E_p[log\tfrac{p(x)}{q(x)}] = \sum_{x\in X} p(x)log\tfrac{p(x)}{q(x)} \\ &= \sum_{x\in X}[p(x)logp(x) - p(x)logq(x)] = \sum_{x\in X}[p(x)logp(x)] - \sum_{x\in X}[p(x)logq(x)] \\ &= -H(p) + E_p[-logq(x)] = H_p(q) - H(p)\end{aligned}$$

Clearly, when p=q, KL divergence equals to 0. In this formula, H_p(q) means that in p(x) distribution, the amount of information required to present x by using q(x) distribution. H(p) means that the entropy of p distribution. Hence, the KL divergence means the additional amount of information required to specify the value of x as a result of using q(x) instead of true distribution p(x).

Cross Entropy and Logistic Regression

Define cross entropy as

$$CE(p,q) = E_p[-logq] = -\sum_{x\in X} p(x)logq(x) = H(p) + D_{KL}(p||q)$$

We can see that H_p(q) in KL divergence formula is actually cross entropy. When p is known, we can view H(p) as a constant, and the cross entropy is equivalent to KL divergence, both represent the similarity of p(x) and q(x). Since p(x) is the true distribution and q(x) is the approximating distribution from our model, our goal is to minimize the distance between these two distributions. Note that it is equivalent to minimize cross entropy and minimize KL divergence. We'll get a minimum point when p=q (KL divergence equals to 0 in this case). It's also called **Principle of Minimum Cross-Entropy** (MCE).

Now back to logistic regression, we have loss function as

$$-logP(y|X) = -(y * log\hat{y} + (1-y) * log(1-\hat{y}))$$

Consider cross entropy in this case

$$\begin{aligned} CE(p,q) &= -\sum_{x\in X} p(x)logq(x) \\ &= -[P_p(x=1)logP_q(x=1) + P_p(x=0)logP_q(x=0)] \\ &= -[plogq + (1-p)log(1-q)] = -[ylog\hat{y} + (1-y)log(1-\hat{y})] \end{aligned}$$

It has the same result as we use maximum likelihood estimation! We can derive the cost function of logistic regression either by MLE or by cross entropy.

Summary

When we want to use a probabilistic model over mutually exclusive classes, we need a way to measure the difference between predicted probabilities ŷ and the ground truth probabilities y. Our goal is to minimize the difference between them. We can see that cross entropy is a reasonable choice for this task. Also, minimizing cross entropy is equivalent to minimizing the negative log likelihood, which we derived from maximum likelihood estimation. Cross entropy is very important and basic concept in probabilistic model. It is also used in softmax function for neural network, which is the most popular technique nowadays.

Congratulations! I hope this article helps a little in understanding what entropy is and how cross entropy relates to logistic regression. As always, I welcome questions, notes, suggestions etc. Enjoy the journey and keep learning!

Machine Learning 30daysofml Entropy Logistic Regression

Follow

Written by Tzu-Chi Lin

138 Followers · Editor for 30 days of Machine Learning

Software Engineer @ Indeed

More from Tzu-Chi Lin and 30 days of Machine Learning