CS281: Advanced ML **October 18, 2017**

Lecture 13: Information Theory

Lecturer: Sasha Rush Scribes: Peter Chang, Ruiqi Chen, Joonhee Choi, Joshua Meier, Raphael Rouvinov, Hyungmok Son

13.1 Announcements

The Midterm is next Monday. The list of topics is on the website. It's open note but not open computer. You can bring your textbook. They'll try to bring copies of the textbook for people who don't have it so they don't have to print out copies.

13.2 Introduction

Interestingly, almost all of information theory is laid out in a single paper, written by Claude Shannon in 1948. We often quote Alan Turing as the 'father' of CS, but for the area we are discussing, that 'father' is Shannon. (Aside: Video of device Shannon built called the Ultimate Machine)

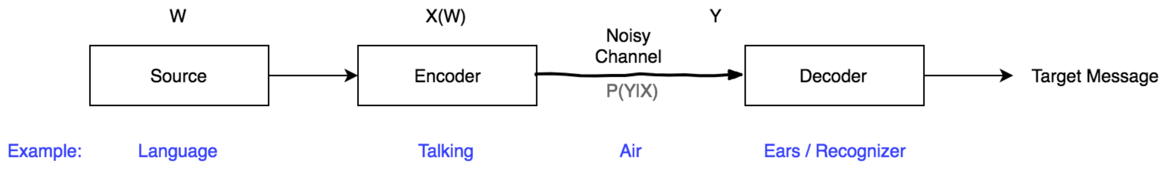
Today, we'll cover the core aspects of information theory and discuss how we'll use it in this class.

Earlier in this class, we've focused on exact methods for inference: MLE, MAP, etc. The one exception was neural networks, which are not convex.

Exact inference is only tractable in a small subset of models. Given most models are intractable, what do we do? We use approximate inference. Here, there are two main types of approximation: optimization-based (which includes coordinate ascent, SGD, and Linear Programming methods) and sample-based. You can use them together, but they have different histories, so we'll discuss them separately.

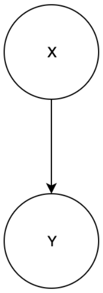
Information theory brings together many of the expectation-maximization methods we've seen earlier into a unified language. In particular, information theory lets us study relative entropy.

13.3 Information theory (Murphy 2.8)



The *Information Source* spits out bits (call this W), which is passed into the *encoder* (giving X(W)). The encoded bits pass through a *noisy channel* giving Y, which is passed into a *decoder* that gives us our target message.

We don't know how the noisy channel will act in a deterministic way. But we have a probability distribution $p(Y|X)$ describing its behavior. The target message also has a distribution $p(x|y) \propto p(x)p(y|x)$. Where $p(x)$ represents the source model, and $p(y|x)$ represents the channel model. We have a simple graphical model here:



Several fields of research are contained in the first diagram. One area of information theory is called channel coding (Noisy Channel in diagram). It studies how to best encode the data so it is most robust to the noisy channel. A second area is called source coding - data compression (Source in diagram). It discusses how we can exploit the way information is naturally represented in the world. For example, compression falls here. (Aside: Hutter prize).

Example: people in speech recognition use this as an analogy for what they are trying to do. The model of what a person wants to say is W. The encoding path is the sound the person makes $X(w)$. The decoder is what we (or the microphone) hear. And the goal, of course, is to uncover the target message.

We can write this analogy:

1. Source: Language
2. Encoder: Talking
3. Noisy Channel: Air
4. Decoder: ears/recognizer

Then, $p(X)$ represents the source model. If you know a person well, you can anticipate what is going on in their head, and you know what they are going to say This makes your $p(X|Y)$ stronger. We multiply this quantity by $p(Y|X)$ which represents the noisiness of the channel.

13.4 Definitions

Entropy:

$$H(X) \overset{\Delta}{=} -\sum_{k=1}^{K} p(x=k)\log_2 p(x=k) = -E_k[\log_2 p(x=k)]$$

For a given random variable X, entropy measures the "uncertainty" in the distribution. Entropy maxes out when we have full uncertainty in the distribution.

13/parabola.png

This comes back to the idea of source coding. If there is full certainty in what can and is sent, then it does not matter what the encoder/decoder does. The answer is trivial. In contrast, if there is full uncertainty, we have to work much harder.

The unit of measure of this is called "bits"($\log_2$) or "shannons" or "nats" ($\log_e$). The average number of bits needed to represent a message is less than

$$H(x) + 1$$

We won't prove this, but this is the fundamental link between information and coding.

Shannon Game In his paper, Shannon proposes the "Shannon Game," which tries to quantify human source coding: given a sequence of text, give a probability distribution over the next letter/word.

THE ROOM WAS NOT VERY LIGHT A SMALL OBLONG READING LAMP ON THE DESK SHED GLOW ON POLISHED ___.

In English, it takes roughly 80 guesses to get the right answer. We write that the perplexity or "how confusing the next prediction is" is

$$2^{H(x)}$$

Minimizing perplexity is where the 'power'/advances of RNNs comes from.

Cross Entropy:

1. p - is a true underlying distribution
2. q - another distribution that approximates p

$$H(p,q) = -\sum_k p(x=k)\log q(x=k) = E_p[\log q(x=k)]$$

Cross entropy tells us expected number of bits to encode true distribution p with q.

We can sample from p to approximate

$$\tilde{x_1}, \ldots, \tilde{x_N} \sim p(x)$$

Then, we compute the minimization of the cross entropy.

$$min_q - \frac{1}{n}\sum_n^N \log q(\tilde{x_n})$$

But this is just the negative log likelihood for categoricals. Last class, we talked about RNNs. We learn q and compare to p. If we can make it more and more like p, this gives us the ability to do compression and source modeling better.

Relative entropy (KL-Divergence):

$$KL(p||q) \triangleq E_p \log \frac{p_(x=k)}{p(q=k)} = \sum_k p_k \log \frac{p_k}{q_k} = \sum_k p_k \log p_k - \sum_k p_k \log q_k = -H(p) + H(p,q)$$

So the relative entropy is the negative entropy of p by itself plus the relative entropy of p and q. It's a way of comparing two distributions, but is not a metric - $KL(p||q) \neq KL(q||p)$

Theorem. $KL(p||q) \geq 0$

Proof. We have that

$$-KL(p||q) = E_p[\log \frac{q_k}{p_k}] \leq \log E_p[\frac{q_k}{p_k}] = \log \sum_k q_k(x) = \log 1 = 0$$

This is only $KL(p||q) = 0$ when $p = q$.

Jensen's Inequality.

$$f(E[x]) \leq E[f(x)]$$

if f is convex. Mostly using when $f = \log$.

Information Geometry (Working with asymmetrical divergence).
We have some p and want $q \in Q$ (set of distributions). There are two options:

1. Forward (moment projection):

$$argmin_{q \in Q} KL(p||q) = -H(p) + H(p,q)$$

Note that $H(p)$ falls out when minimizing q. So instead, we minimize at $H(p,q)$, which is the negative log likelihood. So essentially, we are matching the moments. This equals

$$-E_p \log q_k$$

Issues arise when $q_k \to 0$ and $p_k > 0$. Forward projection will try to avoid zeros.

2. Reverse (information projection):

$$argmin_{q \in Q} KL(q||p) = -H(q) + H(q,p) = -H(q) + E_q \log p_k$$

This method "matches the modes".

Issues arises when $p_k = 0$ and $q_k = 0$. Reverse projection will over predict zeros.

These two approaches fall under a field called Information Geometry. The forward approach is called moment projection. The backward approach is called information projection. "Essentially, these are both methods for computing closeness among different distributions."

Jensen-Shannon divergence – combine the above: add half of the Forward projection value to half of the reverse projection value. This approach is popular right now.

This work is highly related to Generative Adversarial Networks (GANs). This metric is important in producing non-blurry images. Combining two images looks bad, using a method with mode finding yields a better image.

13.5 Demo

We did an example iPython notebook (KL.ipynb) in class.

Next class, we will use the reverse projection to get Q. This is called "variational inference".