Nov. 11, 2021: Lecture 20: continued example of language-model development: latent information; distances between language models

1 Reminder: Motivating example: modeling small-talk vs. non-small talk

1.1 Sample data

Written "vertically" instead of "horizontally" to leave room to write.

Two sequences (in this case, monologue documents):

hi
i
agree
thanks
bye

hi
sell
hi [some stock ticker symbol]
now
thanks

1.2 A skeleton generative story

1. Pick a sentence length ℓ.
2. Pick a sequence of ℓ *states*: where the two possible state types are st for small talk, nst for not small-talk
3. For each state, pick a word according to that state's distribution over single words.

1.3 Ideas for instantiation (these are informal "priors")

1. (from last lecture) st might have a higher probability of *being in longer sentences than in shorter sentences*.
2. (motivation for step 2 and 3 of the generative story) st might have a higher probability of including the word "hi" than nst.
3. (new) st might have a higher probability of starting or ending the sentence than nst.

1.3.1 "Quiz": What is the probability of our first sample-data sequence?

Assume we pick specific lengths (not length "buckets" like "short" vs. "long")

- P(a length-5 sequence (with respect to all possible lengths)) $\times$ P(st nst nst st st) $\times$ P(hi | st) P(i | nst) P(agree | nst) P(thanks | st) P(bye | st)
- P(a length-5 sequence) $\times \sum_{\text{state sequences}\, \sigma_1\sigma_2\sigma_3\sigma 4\sigma_5}$ P(hi | σ_1) P(i | σ_2) P(agree | σ_3) P(thanks | σ_4) P(bye | σ_5)
- P(a length-5 sequence) $\times \sum_{\sigma_1\sigma_2\sigma_3\sigma 4\sigma_5}$ P($\sigma_1\sigma_2\sigma_3\sigma_4\sigma_5$) P(hi | σ_1) P(i | σ_2) P(agree | σ_3) P(thanks | σ_4) P(bye | σ_5)
- Something else

About the discussion of wanting to model the fact that small talk is more likely at the beginning or end of sequences: I've decided talking about transitions vs non-transitions is a red herring.

Instead (and again assuming the sequence length ℓ was already fixed) ...

1. You might consider modeling the choice of ℓ-state sequence to be drawn at random from among all ℓ-state sequences as if there's an 2^ℓ-sided die being thrown. That's 2^ℓ numbers needed, one for each side of the die.

2. Or, you might decide that for each word position, a two-sided coin is flipped to decide whether it's

each individual ℓ-state sequence to be "atomic" (not decomposable) to There are 2^ℓ such numbers involved.

2. Or, you might decide

1. If you think each ℓ-state sequence should be modeled individually with the state history taken into account, there are 2^ℓ such states.

2. But if you think that the state at position i can be considered independent (so you don't have to estimate transition probabilities, since they are position-independent), you get just these states.

$\{\mathsf{st}, \mathsf{nst}\} \times \{1, 2, ..., \ell\}$

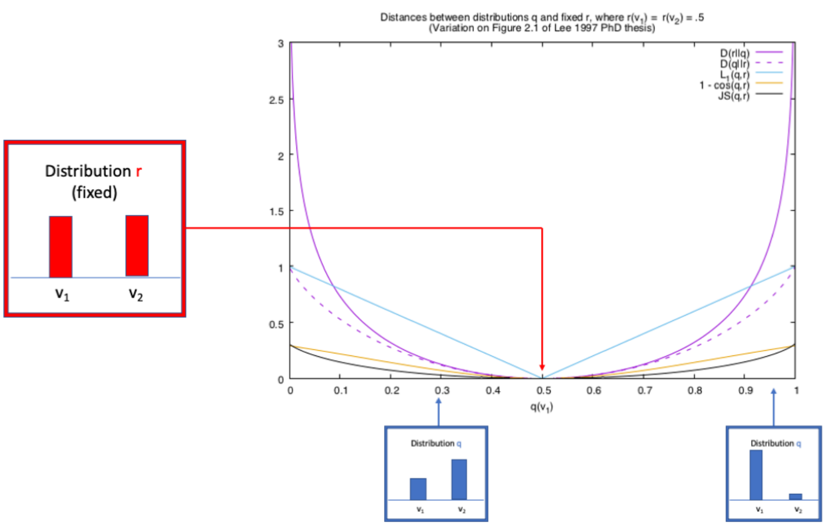
This is 2 $\times \ell$ states, not 2^ℓ, which is a whopping savings in parameters compared to being exponential in ℓ. (When I was talking I thought that there seemed to be too many!)

[There are actually fewer free parameters than $2 \times \ell$; for any position k, if you know $p(statwordk)$, then you already know p(nst at word k), because they sum to one.]

2 Measuring the difference between two "single-word" distributions

We restrict attention to proper distributions $q(\cdot)$ and $r(\cdot)$ over finite "vocabulary" $V = \{v_i\}$. We write q_i and r_i for $q(v_i)$ and $r(v_i)$.

• But LMs give probs to an unbounded number of strings? One can take V to be single words (or whatever), and for a given language model $p(\cdot)$, set p_i to $p(v_i|\text{some context of interest})$ normalized by $\sum_j p(v_j|\text{some context of interest})$.



The *surprisal*[1]:

$$-\log(r_i) = \log \frac{1}{r_i} \tag{1}$$

can be thought of as how *surprised* we should be from the perspective of using r as a model to see v_i, or r's *surprisedness* or *surprisingness* for v_i. The base of the log is customarily taken to be 2, which makes this surprisingness number interpretable as a number of bits of information.[2]

[1] According to Wikipedia, the term was coined in Tribus, 1961, *Thermostatics and Thermodynamics*.

[2] Indeed, a much more common interpretation of equation 1 is as a number of bits needed to encode v_i assuming the distribution r over V.

2.1 Cross-entropy

If we considered the "reference" distribution to be q, then the *cross-entropy*

$$H(q||r) = \sum_i q_i \log \frac{1}{r_i} \tag{2}$$

is the expected surprisedness for r with respect to reference distribution q.[3]

2.2 KL-Divergence

$$D(q||r) = \sum_i q_i \log \frac{q_i}{r_i} \tag{4}$$

2.3 Jensen-Shannon divergence

See Lin, Jianhua. 1991. Divergence measures based on the Shannon entropy. *IEEE Transactions on Information Theory* 37(1): 145-151. Let $\text{avg}_{q,r}$ be the average distribution between q and r.

$$JS(q,r) = \frac{1}{2}\left[D(q||\text{avg}_{q,r}) + D(r||\text{avg}_{q,r})\right] \tag{5}$$

2.4 Skew divergence

See Lee, Lillian. 1999. Measures of distributional similarity. In *Proceedings of the ACL*, 25-32.

$$\text{skew}_\beta(q||r) = D(q||\beta \cdot r + (1-\beta)q) \tag{6}$$

Values used include $\beta = .99$.

[3] *How you often see this in papers:* If the "reference" distribution is taken to be the one induced from the empirical counts from a sample $S = w_1 w_2 \ldots$, where each $w_k \in V$ and the length of the sample is L, then this can be refactored as:

$$\hat{H}_S(r) = \frac{1}{L}\sum_{k=1}^{L} \log \frac{1}{r(w_k)} \tag{3}$$