CS109 Lecture 18: LLM Learning Guide

Information Theory

Summer 2026

Lecture 18: LLM Learning Guide

Before lecture: read the Lecture 18 slides (information theory). Then open your favorite LLM and work through the concepts below **in order**. Each concept has a *Learn* prompt and a *Test me* prompt. Attempt the "Test me" questions yourself before asking the LLM to grade you.

How to get the most out of this:

- Tell the LLM your level: "I'm a student in Stanford's CS109 (probability for computer scientists). Explain at that level."
- After any answer, ask "why?" at least once and ask it to show each step.
- When it states a formula, ask it to define every symbol before continuing.
- Attempt practice yourself first, then say: "Here is my reasoning: _____. Where exactly am I wrong, if anywhere?"

Concept 1: Surprise (information content)

Learn: > "Explain the concept of the surprise (aka information content or self-information) of an event: $I(E) = \log_2 \frac{1}{P(E)}$, measured in bits. Explain the desiderata that lead to this formula: high-probability events should be unsurprising, low-probability events surprising, the relationship monotonic, and surprise should *add* across independent events. Show why the log is what makes additivity work, and compute a few examples."

Test me: > "Give me several events with probabilities and make me compute their surprise in bits. Include a probability-1 event and a pair of independent events, and check that I can prove $I(E \cap F) = I(E) + I(F)$ for independent events."

Concept 2: Entropy as expected surprise

Learn: > "Define the entropy of a random variable as its expected surprise: $H(X) = \sum_x P(X{=}x) \log_2 \frac{1}{P(X=x)}$. Walk through the derivation from 'expectation of $I(X{=}x)$' using log rules, explain why entropy measures uncertainty (a point-mass distribution has $H = 0$; a uniform distribution over k values has $H = \log_2 k$, the maximum), and explain the convention for handling zero-probability values."

Test me: > "Give me 3 or 4 small PMFs (including a fair coin, a heavily biased coin, and a non-uniform 4-value distribution) and make me compute the entropy of each by hand and rank them by uncertainty before computing. Check whether my ranking intuition matched the math."

Concept 3: Information gain — choosing the best question

Learn: > "Explain how to choose the most informative question (as in twenty-questions, decision trees, or Wordle): each possible answer restricts the distribution of the unknown X, leaving a conditional distribution with its own entropy; since the answer is random, we score a question by its *expected remaining entropy*, weighted by the probability of each answer. Information gain is $H(X)$ minus that expectation. Work an example with a *non-uniform* prior, and show that the 'split the possibilities in half' heuristic can be beaten when the prior is skewed."

Test me: > "Set up a guessing game where X has 4–5 possible values with a non-uniform prior, and offer me two candidate yes/no questions. Make me compute the expected remaining entropy of each and pick the better question. Then ask me why we don't just score a question by its best-case answer."

Concept 4: Entropy in code

Learn: > "Show a short Python function that computes the entropy of a distribution represented as a dictionary PMF (loop over values, skip zero probabilities, accumulate $p \log_2 \frac{1}{p}$). Then show how to compute an *expected* remaining entropy over the answers to a question, and explain how this becomes the core loop of a decision-tree learner or a Wordle solver that picks the highest-information-gain guess."

Test me: > "Give me a small PMF dictionary and make me trace an entropy function by hand, including why zero-probability entries are skipped. Then give me two candidate 'questions' as partitions of the values and make me write 5 lines of pseudocode that scores them by expected remaining entropy."

Concept 5: KL divergence

Learn: > "Explain Kullback-Leibler divergence as the expected *excess* surprise from using a model distribution Q when reality is P: $D(P\|Q) = \sum_x P(x) \log \frac{P(x)}{Q(x)}$. Walk through the derivation from 'surprise under Q minus surprise under P, in expectation under P'. Explain its key properties: $D \geq 0$, zero iff $P = Q$, and *not symmetric* (so not a true distance). Work one small discrete example both directions to show the asymmetry."

Test me: > "Give me two small discrete distributions (like two biased coins or two 3-value PMFs) and make me compute $D(P\|Q)$ and $D(Q\|P)$ by hand. Ask me to interpret each direction in words (which one is the model and which is reality), and to explain when the divergence blows up to infinity."

Concept 6: Comparing distributions in the wild

Learn: > "Briefly survey the three ways from CS109 to score how similar two distributions are — total variation distance, earth mover's distance, and KL divergence — with the intuition for each and one situation where each is the natural choice. Then explain how KL divergence is used in practice: scoring how well a predicted distribution (like a Poisson model of hurricane counts) matches observed data, and as the loss function idea behind cross-entropy in machine learning."

Test me: > "Describe two or three scenarios where I must compare a model distribution to data (a Poisson fit, a weather forecast, a language model's next-token distribution) and ask me which comparison measure I'd use and in which direction I'd compute a KL divergence, and why. Push back on my answers with follow-up questions."

Wrap-up prompt (after all six concepts)

> "Give me one multi-part CS109-style problem in which I must (1) compute the surprise of some events, (2) compute and compare the entropy of two PMFs, (3) choose between two questions by

expected remaining entropy under a non-uniform prior, and (4) compute a small KL divergence in both directions and interpret the asymmetry. Let me solve every part, then grade each part, tell me which concept it tested, and tell me the single concept I should review most."

Note: entropy and KL divergence power decision trees, Wordle solvers, adaptive testing, data compression, and the cross-entropy loss you will meet again in logistic regression and deep learning later in this course.

Bring any sticking points to lecture; the TAs and instructor will help you work through them on the worksheet.