CS109 Lecture 25: LLM Learning Guide

Regression and Reinforcement Learning

Summer 2026

Lecture 25: LLM Learning Guide

Before lecture: read the Lecture 25 slides (regression and reinforcement learning). 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?"

A warning specific to today: linear regression is the most over-explained topic on the internet, and almost all of that writing starts from "minimize the sum of squared errors" as if it were a definition. In CS109 it is a *conclusion.* If the LLM opens with least squares, stop it and say: "Start from a probabilistic assumption about $Y \mid X = x$ and derive least squares from maximum likelihood." Likewise, most reinforcement learning material will drown you in Q-learning and policy gradients — for CS109 you want expectation, geometric series, and the Beta distribution, not the algorithm zoo.

Concept 1: The five machine learning tasks

Learn: > "Lay out the landscape of supervised and unsupervised machine learning tasks — binary classification, multi-class classification, regression, reinforcement learning, and generation — and for each one tell me what the output random variable Y is, what distribution we assume for $Y \mid X = x$, and what function we apply to $\theta^T x$ to get the parameter of that distribution. Make a table. Then explain why generation is not really a separate model type: an LLM's output layer is a Categorical over the vocabulary, and generation is sampling from it in a loop. Finally, explain what makes reinforcement learning structurally different from the other four."

Test me: > "Give me eight concrete prediction problems drawn from different domains and make me classify each one by task type, name the output random variable and its support, and state the assumed conditional distribution. Include at least one that is ambiguous between regression and multi-class classification, and make me argue both sides."

Concept 2: The linear regression model and its assumptions

Learn: > "Set up linear regression the CS109 way. Start from $y^{(i)} = \theta^T x^{(i)} + Z^{(i)}$ with $Z^{(i)} \sim N(0, \sigma^2)$ i.i.d., and show me carefully — using the fact that adding a constant to a Normal shifts its mean and leaves its variance alone — that this is exactly the statement $Y^{(i)} \mid X = x^{(i)} \sim N(\theta^T x^{(i)}, \sigma^2)$. Then enumerate every assumption hiding in that one line: linearity of the conditional mean, Normality of the errors, independence, and constant variance. For each assumption, give me a realistic dataset where it fails and describe what goes wrong. Explain why assuming the noise has mean zero costs us nothing (the intercept absorbs any constant offset). Finally, put logistic regression and linear regression side by side: same recipe, different assumed distribution, different function applied to $\theta^T x$ (sigmoid versus identity)."

Test me: > "Quiz me on the linear regression assumptions: give me described datasets and make me identify which assumption is violated and what the practical consequence is. Then make me state the conditional distribution, the log-likelihood of a single datapoint, and the range of the prediction for both linear and logistic regression, from memory, and grade me."

Concept 3: Deriving least squares from maximum likelihood

Learn: > "Derive the linear regression objective from scratch as maximum likelihood estimation. Step 1: state the assumption $Y \mid X = x \sim N(\theta^T x, \sigma^2)$. Step 2: write the likelihood as a product of Normal PDFs over the i.i.d. data, take the log, and simplify to >

$$LL(\theta) = \sum_{i=1}^{n} \log\left[\frac{1}{\sigma\sqrt{2\pi}}\right] - \frac{1}{2}\left(\frac{y^{(i)} - \theta^T x^{(i)}}{\sigma}\right)^2.$$

> Step 3: argue carefully why the constant term and the positive factor $\frac{1}{2\sigma^2}$ can both be dropped without changing the argmax, and conclude that maximizing the likelihood is *identical* to minimizing the sum of squared errors. Emphasize that least squares is a consequence of the Gaussian assumption, not a definition. Then take the partial derivative with respect to θ_j, showing exactly where the chain rule is used and how the two minus signs cancel, to get $\sum_i 2(y^{(i)} - \theta^T x^{(i)}) x_j^{(i)}$, and write the gradient ascent update rule."

Test me: > "Make me reproduce the entire derivation with no hints, grading each step and catching me if I drop a constant illegally or lose a sign. Then ask me why we are allowed to discard σ from the argmax over θ even though σ is genuinely a parameter of the model, and ask me what would change if σ differed from datapoint to datapoint."

Concept 4: Computing with the model

Learn: > "Walk me through a complete worked example of gradient ascent for linear regression on a tiny dataset with one feature plus a bias term. Show the predictions, the errors, both partial derivatives, one parameter update with a stated learning rate, and the sum of squared errors before and after. Then show me that at the optimum the residuals sum to zero, and explain why: the $j = 0$ component of the gradient is $\sum_i (y^{(i)} - \hat{y}^{(i)})$ because $x_0 = 1$, and it must be zero at a maximum. Then explain what the noise variance σ^2 is for: give me the MLE $\hat{\sigma}^2 = \frac{1}{n}\sum_i (y^{(i)} - \hat{y}^{(i)})^2$, point out that it equals the minimized SSE divided by n, and show how it turns a point prediction into a full predictive distribution $N(\theta^T x, \hat{\sigma}^2)$ that you can use to compute probabilities and intervals. Explain why $\hat{\sigma}^2$ must be estimated on a test set, not the training set."

Test me: > "Give me a small labelled dataset and a starting θ and make me do everything by hand: predictions, errors, SSE, both partials, one gradient ascent step, and the new SSE. Then give me a list of test-set residuals and make me estimate σ^2, compute the probability that a new prediction is off by more than a given amount, and give a 95% predictive interval. Check my arithmetic and my use of the Normal table."

Concept 5: The same gradient, every time

Learn: > "Put four gradients next to each other: the Bernoulli MLE, logistic regression $\sum_i [y^{(i)} - \sigma(\theta^T x^{(i)})] x_j^{(i)}$, the neural network output layer $[y - \hat{y}] h_j$, and linear regression $\sum_i 2[y^{(i)} - \hat{y}^{(i)}] x_j^{(i)}$. Point out that they are all (observed − expected) × (input), up to a constant that gets absorbed into the learning rate. Explain at a CS109 level why this is structural rather than coincidental: sigmoid and the identity function are the canonical link functions for the Bernoulli and the Normal, both members of the exponential family, and for a canonical link the derivative of the log-likelihood with respect to the natural parameter is always the difference between what you saw and what you expected. Keep it conceptual — I do not need exponential family vocabulary for the exam, I need to know that the pattern is not luck."

Test me: > "Give me a model I have never seen — for example, Poisson regression with $Y \mid X = x \sim \text{Poi}(e^{\theta^T x})$ — and make me guess the form of its gradient before deriving it, then make me actually derive it and check whether my guess was right. Then ask me what a single line of gradient ascent code would have to change to switch between logistic, linear, and Poisson regression."

Concept 6: Reinforcement learning with CS109 tools

Learn: > "Explain reinforcement learning at the level of CS109 probability, not at the level of a deep RL course. Cover: (1) the setup — an agent takes actions, the environment returns rewards, and there are no labels anywhere; (2) the discounted return $G = \sum_{t=0}^{\infty} \gamma^t R_t$, why the geometric series converges for $\gamma < 1$, what goes wrong at $\gamma = 1$, and how to compute $E[G]$ using linearity of expectation; (3) the credit assignment problem, where a reward arrives long after the action that earned it; (4) why the i.i.d. assumption we have used all quarter breaks down, since the agent's own policy determines what data it sees next; and (5) the exploration/exploitation tradeoff, framed as a multi-armed bandit where each arm's reward probability has a Beta posterior updated from Bernoulli observations. Show me how Thompson sampling — sample one θ from each arm's posterior, pull the arm with the largest sample — solves exploration using nothing but the Beta-Bernoulli conjugacy from Lecture 14. Do not go into Q-learning or policy gradients."

Test me: > "Give me a bandit scenario with two or three options, each with a stated number of successes and failures starting from a Beta(1,1) prior. Make me compute each posterior, its mean, and its standard deviation, then decide what a greedy agent would do and explain why that could be wrong. Then make me compute an expected discounted return for a given γ and reward process. Finally, quiz me on what distinguishes reinforcement learning from supervised learning and push back if I say 'the reward is just the label.' "

Wrap-up prompt (after all six concepts)

> "Give me one multi-part CS109-style problem in which I must (1) state the linear regression probability assumption and justify it, (2) derive least squares from the Gaussian log-likelihood and take the partial derivative, (3) run one step of gradient ascent by hand on a small dataset and report the SSE before and after, (4) estimate the noise variance from test residuals and use it to compute a probability and a predictive interval, and (5) compute Beta posteriors for a two-armed bandit and reason about exploration versus exploitation. 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."

A thought to carry into lecture: we have now written down four different models this quarter — Bernoulli, Poisson, logistic, and linear — and every single one was built the same way. Assume a distribution. Write the likelihood of the data. Take the log. Take the derivative. Walk uphill. The models differ only in step one. If you understand that sentence, you understand the entire second half of CS109.

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