CSE Ph.D. Qualifying Exam, Fall 2025: Data Analysis

Instructions:

- This is a CLOSED BOOK exam. No books or notes are allowed.
- Please answer three of the following four questions. All questions are graded on a scale of 10. If you answer all four, all answers will be graded and the three lowest scores will be used in computing your total.
- Please write clearly and concisely, explain your reasoning, and show all work. Points will be awarded for clarity as well as correctness.
- Good luck!

Q1: Generative Models

Consider Alice and Bob are asked to fit a generative model $p_\theta(x)$ over a given dataset $\mathcal{D} = \{x_i\}_{i=1}^n$ with $x_i \in \mathbb{R}^d$, where θ denotes the parameters of the model.

(a) [5 points] Alice considered a simple Gaussian distribution:

$$p_\theta(x) = \frac{1}{\sqrt{(2\pi)^d \det(\Sigma)}} \exp\left(-\frac{1}{2}(x-\mu)^\top \Sigma (x-\mu)\right),$$

with $\theta := (\mu, \Sigma)$.

(i) [2 points] Please derive the maximum log-likelihood estimator (MLE) of the Gaussian model $p_\theta(x)$ over given data $\mathcal{D}$.

(ii) [3 points] Please derive the gradient of MLE w.r.t. Σ and μ, and the optimal closed-form solution.

(b) [5 points] Bob considers the generalization of Gaussian distribution, which is known as *energy-based model (EBM)*:

$$p_\theta(x) := \frac{1}{Z_\theta} \exp\big(f_\theta(x)\big), \quad \text{with} \quad Z_\theta := \int \exp\big(f_\theta(x)\big) dx.$$

(i) [3 points] Please derive the maximum log-likelihood estimator (MLE) of EBM $p_\theta(x)$ over given data $\mathcal{D}$.

(ii) [2 points] Please derive the gradient of the MLE of EBM, *i.e.*,

$$\nabla_\theta \widehat{\mathbb{E}}_{x\sim\mathcal{D}}\left[\log p_\theta(x)\right] = \widehat{\mathbb{E}}_{x\sim\mathcal{D}}\left[\nabla_\theta f_\theta(x)\right] - \mathbb{E}_{x\sim p_\theta(x)}\left[\nabla_\theta f_\theta(x)\right]. \tag{1}$$

Q2: EM for Mixture of Multivariate Bernoulli Distributions

Suppose we observe binary vectors $x \in \{0,1\}^D$ generated from a mixture of K multivariate Bernoulli distributions:

$$p(x) = \sum_{k=1}^K \pi_k \,\mathrm{Bern}(x \mid \theta_k), \tag{2}$$

where π_k is the mixing weight for the k-th component ($0 \le \pi_k \le 1$, $\sum_{k=1}^K \pi_k = 1$), and

$$\mathrm{Bern}(x \mid \theta_k) = \prod_{d=1}^D \theta_{kd}^{x_d} (1-\theta_{kd})^{1-x_d}. \tag{3}$$

We now re-express the model using explicit latent indicator variables $z \in \{e_1, e_2, \ldots, e_K\}$, where e_k is the k-th standard basis vector in $\mathbb{R}^K$. Equivalently, $z = (z_1, \ldots, z_K)$ with $z_k \in \{0,1\}$ and $\sum_{k=1}^K z_k = 1$.

(a) [2 points] **Latent Variable Representation:** Using the latent variable representation, show that

$$p(x) = \sum_z p(z)\, p(x \mid z)$$

reduces to the original mixture form Eq. (2). Clearly define $p(z)$ and $p(x \mid z)$ in this context. *Hint:* $p(z)$ follows a categorical distribution over K categories. The general density form of a categorical distribution is $p(a) = \prod_{k=1}^{K} \pi_k^{a_k}$.

(b) [3 points] **E-step Posterior Derivation:** In the E-step of EM, we compute the posterior responsibility for component k:

$$\gamma_{ik} \triangleq p(z_k = 1 \mid x_i).$$

Derive a formula for γ_{ik} in terms of the fixed parameters π_k and θ_k using Bayes' rule. Interpret γ_{ik} in words.

(c) [3 points] **M-step Parameter Updates:** In the M-step, we re-estimate π_k and θ_k to maximize the expected complete-data log-likelihood:

$$\ell(\pi, \theta) = \sum_{i=1}^{N} \sum_{k=1}^{K} \gamma_{ik} \log \left[\pi_k \prod_{d=1}^{D} \theta_{kd}^{x_{id}} (1 - \theta_{kd})^{1 - x_{id}} \right].$$

Derive closed-form update formulas for π_k and each θ_{kd}. Use a Lagrange multiplier to enforce the mixing weight constraints.

(d) [2 points] **Connection to K-Means:** Describe how K-Means clustering can be interpreted as a limiting case of EM for certain discrete mixture models. Briefly outline the "E-step" and "M-step" analogs in K-Means.

Q3: Decision Trees and Random Forests

After desperately solving all the density estimation problems, Alice and Bob are now asked to do a supervised learning problem by fitting a labeled-dataset $\mathcal{D} = \{x_i, y_i\}_{i=1}^n$ using decision trees.

(a) [2 points] Alice wants to proceed with the first step of decision trees: splitting! However, Alice does not know how to properly quantify the quality of a split. Can you help Alice define the "split" and the metric used to quantify the quality of a split in decision trees?

(b) [1 point] After defining the quality of a split, Bob is trying to help select the best split to split the dataset into two subsets. Please help Bob write down the splitting optimization problem.

(c) [1 point] Following the above question, what is the computation cost of finding the best split? Please write the computation cost in the big-O notation in terms of n, d and explain.

(d) [2 points] Alice is ready to implement the decision tree classification algorithm. Please help write the pseudocode for running decision trees on dataset $\mathcal{D}$ for Alice.

(e) [2 points] Please help Alice analyze the computation cost of the decision tree algorithm. Please write the computation cost in the big-O notation in terms of n, d and explain. Let us suppose the tree is balanced, i.e., the depth is roughly $O(\log n)$.

(f) [2 points] Bob would like to further improve the decision tree performance by using random forest. Please describe how random forest works in classification problem. Please also describe how you would avoid the overfitting issue.

Q4: Why Does Averaging Language Models Help?

A (next-token) *language model* $p(x \mid h)$ maps a history h (context) to a probability distribution over the next token $x \in \mathcal{V}$. Suppose we have M trained language models $\{p_m\}_{m=1}^{M}$. Their *uniform ensemble (mixture)* is

$$\bar{p}(x \mid h) \;=\; \frac{1}{M} \sum_{m=1}^{M} p_m(x \mid h).$$

Given a dataset of contexts and realized next tokens $\mathcal{D} = \{(h_i, x_i)\}_{i=1}^{N}$, define the average *negative log-likelihood (NLL)* and *perplexity (PPL)*:

$$\mathrm{NLL}(p) \;:=\; -\frac{1}{N} \sum_{i=1}^{N} \log p(x_i \mid h_i), \quad \mathrm{PPL}(p) \;:=\; \exp\big(\mathrm{NLL}(p)\big).$$

(a) [4 points] Ensembling reduces NLL on average (hence lowers perplexity). To prove this, show that

$$\mathrm{NLL}(\bar{p}) \;\le\; \frac{1}{M} \sum_{m=1}^{M} \mathrm{NLL}(p_m) \quad \Longrightarrow \quad \mathrm{PPL}(\bar{p}) \;\le\; \Big(\prod_{m=1}^{M} \mathrm{PPL}(p_m) \Big)^{1/M}.$$

Remarks: This inequality shows that averaging models in *probability space* cannot do worse than the *average* component on NLL; equivalently, the ensemble's perplexity is no larger than the *geometric mean* of component perplexities.

(b) [2 points] When is there no gain? State the equality conditions for part (a).

(c) [4 points] The ensemble is not far from the best single model (log-sum-exp bound). To see this, prove

$$\mathrm{NLL}(\bar{p}) \;\le\; \min_{1 \le m \le M} \mathrm{NLL}(p_m) \;+\; \log M.$$

Remarks: This inequality states that even if ensembling doesn't beat the very best component, its NLL is within at most $\log M$ nats per token (i.e., $\log_2 M$ bits) of the best model.