EE660 Homework 4

Assigned: 10/28/25, Due: 11/13/25

Instructions: You may collaborate with others on this problem set, but each student must independently write their own solutions. We highly encourage you to use LaTeX to typeset your solutions. We will accept handwritten assignments, however if the solution to a problem is too illegible for the grader to read, then they may use their discretion and consider the problem incomplete. Solutions are due by 11:59pm Pacific Time on the due date, and are only to be submitted on DEN. Do not email the course staff with your assignment.

Template: The link `https://www.overleaf.com/read/hjgknqqhryqy` contains a basic LaTeX template that you may use. Note, however, that you are not required to use this template.

1. Problem 1

A function $f : \mathbb{R}^d \mapsto \mathbb{R}$ is *convex* if the following inequality holds:

$$f(\alpha x + (1-\alpha)y) \leqslant \alpha f(x) + (1-\alpha)f(y), \quad x, y \in \mathbb{R}^d,\ \alpha \in [0,1].$$

In this problem, we will consider the function:

$$f(x) := \frac{1}{2}\|x - 0.5\sin(x)\|^2,$$

where $\sin(x)$ of a vector is applied coordinate-wise, i.e., $\sin(x) = (\sin(x_1), \ldots \sin(x_d))$.

(a) Show that f is *not* convex.

(b) Show that $f_\star = \inf_{x\in\mathbb{R}^d} f(x) = 0$.

(c) Show that f is 1/4-PL, i.e., for all $x \in \mathbb{R}^d$,

$$\|\nabla f(x)\|^2 \geqslant \frac{1}{2}(f(x) - f_\star).$$

(d) Now consider the generalized function

$$f_\varepsilon(x) := \frac{1}{2}\|x - \varepsilon\sin(x)\|^2, \quad \varepsilon \in \mathbb{R}.$$

Prove or disprove the following statement: for every $\varepsilon \in \mathbb{R}$, the function f_ε is μ-PL (with $\mu > 0$ possibly depending on the specific value of ε).

2. Problem 2

Let $x_1, \ldots, x_n$ be drawn iid from a distribution $p(x)$ over X, and let $\mathscr{P}$ be a family of distributions over X indexed by a finite set $\mathscr{F}$:

$$\mathscr{P} = \{p_f(x) \mid f \in \mathscr{F}\}.$$

Suppose that there exists a $B > 0$ such that the following boundedness condition holds uniformly over $\mathscr{P}$:

$$\max_{f \in \mathscr{F}} \sup_{x \in \mathsf{X}} \left| \log \frac{p(x)}{p_f(x)} \right| \leqslant B.$$

Let $\hat{p}_f \in \mathscr{P}$ denote maximum likelihood solution:

$$\hat{p}_f \in \arg\max_{f \in \mathscr{F}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \log p_f(x_i) \right\}.$$

Show that with probability at least $1 - \delta$ over the randomness of the $\{x_i\}$'s,

$$\mathrm{KL}(p \parallel \hat{p}_f) \leqslant \min_{f \in \mathscr{F}} \mathrm{KL}(p \parallel p_f) + 2B\sqrt{\frac{2\log(2|\mathscr{F}|/\delta)}{n}}.$$

Hint: This is an ERM problem in disguise; use the tools from the first part of the course (before the midterm) to analyze it.

3. Problem 3

Let P denote a distribution over $\mathbb{R}^d$ such that both the mean $\mu_P := \mathbb{E}_{X \sim P}[X]$ and covariance $\Sigma_P := \mathbb{E}_{X \sim P}[(X - \mu_P)(X - \mu_P)^{\mathsf{T}}]$ exist, and the covariance Σ_P is positive definite. Consider the following Gaussian *forward KL-projection*:

$$(\hat{\mu}, \hat{\Sigma}) = \arg\min_{\mu \in \mathbb{R}^d, \Sigma \in \mathrm{Sym}_+^d} \mathrm{KL}(P \parallel \mathsf{N}(\mu, \Sigma)),$$

where Sym_+^d is the space of $d \times d$ positive definite matrices. Show that the solution to the Gaussian forward KL-projection is given by *moment matching*, i.e., $\hat{\mu} = \mu_P$ and $\hat{\Sigma} = \Sigma_P$.

Note: You may assume that the distribution P has a valid density function $p(x)$ on $\mathbb{R}^d$ which is positive everywhere, so that the KL-divergence $\mathrm{KL}(P \parallel \mathcal{N}(\mu, \Sigma))$ is well-defined.

Hint: You may use the following statement without proof. Given a positive definite $V \in \mathrm{Sym}_+^d$:

$$V = \arg\min_{\Sigma \in \mathrm{Sym}_+^d} \left\{ \log\det(\Sigma) + \mathrm{tr}(\Sigma^{-1} V) \right\}.$$

4. Problem 4

For $\mu \in [0,1]^d$, consider the following Bernoulli distribution:

$$\phi(x;\mu) := \prod_{\ell=1}^{d} \mu_\ell^{x_\ell}(1-\mu_\ell)^{1-x_\ell}, \quad x \in \{0,1\}^d.$$

Here, x_ℓ denotes the ℓ-th index of x (and likewise μ_ℓ denotes the ℓ-th index of μ). Consider the following k-mixture model for x:

$$z \sim \mathrm{Cat}(\pi_1, \ldots, \pi_k), \quad x \mid z \sim \phi(x;\mu_z),$$

where $\{\pi_j\}_{j=1}^k$ is an element of the $(k-1)$-dimensional simplex in $\mathbb{R}^k$, and $\{\mu_j\}_{j=1}^k \subset [0,1]^d$ are k separate cluster parameters. Let $\theta = (\{\pi_j\}_{j=1}^k, \{\mu_j\}_{j=1}^k)$ denote the parameters of the mixture model. For an observation $x \in \{0,1\}^d$, mixture parameters θ, and cluster index $j \in \{1,\ldots,k\}$, define:

$$r_\theta(j;x) := \mathbb{P}_\theta(z{=}j \mid x) = \frac{\pi_j \phi(x;\mu_j)}{\sum_{j'=1}^k \pi_{j'}\phi(x;\mu_{j'})}.$$

Furthermore, given n samples $x_1, \ldots, x_n \in \{0,1\}^d$, define:

$$r_\theta(j) := \sum_{i=1}^{n} r_\theta(j;x_i).$$

Show that, given parameters $\theta_t = (\{\pi_j^{(t)}\}_{j=1}^k, \{\mu_j^{(t)}\}_{j=1}^k)$, the EM algorithm updates the mixture parameters to $\theta_{t+1} = (\{\pi_j^{(t+1)}\}_{j=1}^k, \{\mu_j^{(t+1)}\}_{j=1}^k)$ via the following update rule for each $j \in \{1,\ldots,k\}$:

$$\pi_j^{(t+1)} = \frac{r_{\theta_t}(j)}{n}, \quad \mu_j^{(t+1)} = \sum_{i=1}^{n} \frac{r_{\theta_t}(j;x_i)}{r_{\theta_t}(j)} x_i.$$

5. Problem 5

Implement the Bernoulli mixture EM algorithm described in Problem 4 on binarized MNIST. Specifically, perform the following steps:

1. Download and *binarize* the MNIST train and test dataset. The easiest way to do this is to use either `tensorflow_datasets`[1] or `torchvision.datasets`.[2] *Binarizing* refers to taking the raw pixel values and thresholding them to 1 if the pixel value is $\geqslant 128$ and 0 otherwise.

2. Implement and run the Bernoulli mixture EM algorithm described in Problem 4 on the MNIST training dataset, using $k = 15$ clusters. You should not need to run more than 100 iterations of EM for it to converge.

[1] `https://www.tensorflow.org/datasets/catalog/mnist`
[2] `https://pytorch.org/vision/main/datasets.html`

3. Provide the following plots: (a) a plot showing the log-likelihood of both the train and test datasets, as a function of the EM iteration count, (b) a plot visualizing each of the learned clusters after convergence, and (c) a plot showing, for each cluster, a histogram of the true labels that were "assigned" to that cluster. "Assigned" is used in quotation since we know that EM does not perform hard assignments. However, we can force a hard assignment by taking $\arg\max_{j \in \{1,\ldots,k\}} r_{\theta_T}(j; x)$ as the cluster assignment for observation x, where θ_T is the EM parameters of the final iteration. For plot (c), plot both the histograms of the training data and the test data.

You may implement Problem 5 in any programming language / scientific computing library you prefer. However, skeleton code of a `jax`[3] implementation is provided in `hw4_prob5_scaffolding.ipynb`. In addition to providing the plots in your writeup, we ask that you upload your code for Problem 5 on DEN.

Here are the folllowing general characteristics you should expect out of the plots:

(a) Plot (a) should monotonically increase for the training log probability, as we proved that EM satisfies this property. Also, we found that the test log probability does a fairly good job of tracking the train log probability.

(b) For plot (b), your cluster visualizations should generally look like digits. Since there are only 10 true clusters, however, you should expect that some of your clusters are either duplicates or some mix of digits (e.g., blending 3 and 8 together).

(c) For plot (c), for the clusters which truly look like digits, you should expect the histograms to be very concentrated around the corresponding digit.

A final warning regarding numerical issues in the implementation: For a given cluster mean μ, if any of the coordinates of μ are very close to zero or one, then $\phi(x; \mu)$ is approximately zero for certain x, or equivalently $\log \phi(x; \mu)$ approaches $-\infty$. This can easily cause your EM updates to start generating `NaN`s. The easily way to avoid these issues is to clip the updated EM parameters away from zero or one.

[3] `https://github.com/google/jax`