STATS 305B: Practice Final

Write your name here:

Instructions:

- Write on the exam. We will scan it. If you use the extra pages at the back, label clearly.
- You can bring handwritten notes on **two sides** of an 8.5x11" piece of paper.
- Unless otherwise specified, you can write your answers using the "named distribution PDF shorthand," e.g. write the pdf of a Gaussian distribution with mean μ and variance σ^2 as $\mathcal{N}(x; \mu, \sigma^2)$.

Some tips:

1. It's usually a good idea to look through the whole exam before taking it to make sure there aren't missing pages; and so that you roughly know what you are up against
2. It's usually a good idea to skip questions if you are stuck and circle back.

Stanford Honor Code

1. The Honor Code is an undertaking of the students, individually and collectively:
   - that they will not give or receive aid in examinations; that they will not give or receive unpermitted aid in class work, in the preparation of reports, or in any other work that is to be used by the instructor as the basis of grading;
   - that they will do their share and take an active part in seeing to it that others as well as themselves uphold the spirit and letter of the Honor Code.
2. The faculty on its part manifests its confidence in the honor of its students by refraining from proctoring examinations and from taking unusual and unreasonable precautions to prevent the forms of dishonesty mentioned above. The faculty will also avoid, as far as practicable, academic procedures that create temptations to violate the Honor Code.
3. While the faculty alone has the right and obligation to set academic requirements, the students and faculty will work together to establish optimal conditions for honorable academic work.

Problem 1: *Mixture Models*

Consider a problem of predicting $y \in \mathbb{R}$ given inputs $\boldsymbol{x} \in \mathbb{R}^D$. Suppose we believe the targets are a linear function of the input on some region of input space and a different linear function on another region. We can encode these beliefs using a latent variable. The resulting model is called a *mixture of experts* model.

In this model, an "expert" $z \in \{0,1\}$ is selected based on the input $\boldsymbol{x}$, then the target y is generated as a linear function of $\boldsymbol{x}$, where the linear parameters depend on which expert was chosen. More formally, we use the following probabilistic model:

$$p(z_n = 1 \mid \boldsymbol{x}_n; \boldsymbol{\theta}) = \sigma(\boldsymbol{c}^\top \boldsymbol{x}_n)$$
$$p(y_n \mid \boldsymbol{x}_n, z_n; \boldsymbol{\theta}) = \begin{cases} \mathcal{N}(y_n \mid \boldsymbol{w}_0^\top \boldsymbol{x}_n, \sigma_0^2), & \text{if } z_n = 0 \\ \mathcal{N}(y_n \mid \boldsymbol{w}_1^\top \boldsymbol{x}_n, \sigma_1^2), & \text{if } z_n = 1 \end{cases}$$

The parameters of this model are $\boldsymbol{\theta} = \{\boldsymbol{c}, \boldsymbol{w}_0, \boldsymbol{w}_1, \sigma_0, \sigma_1\}$. Suppose we observe data $\{(\boldsymbol{x}_n, y_n)\}_{n=1}^N$. Let $\boldsymbol{X} \in \mathbb{R}^{N \times D}$ denote the matrix of inputs with rows $\boldsymbol{x}_n^\top$, let $\boldsymbol{y} \in \mathbb{R}^N$ denote the vector of targets, and let $\boldsymbol{z} \in \{0,1\}^N$ denote the vector of latent variables. In this question, you will derive the EM updates for this model.

(a) As a warm-up, is the following statement true or false? *When applying the EM updates to the parameters, they may get trapped in a local optima.*

(b) Write the joint log-probability $\log p(\boldsymbol{y}, \boldsymbol{z} \mid \boldsymbol{X}; \boldsymbol{\theta})$ for this model. Do not replace the sigmoid with its definition $\sigma(x) = \frac{1}{1+\exp(-x)}$ or substitute the pdf of the normal distribution.

Hint: Use that:

$$p(z_n \mid \boldsymbol{x}_n; \boldsymbol{\theta}) = \left[\sigma(\boldsymbol{c}^\top \boldsymbol{x}_n)\right]^{z_n} \left[(1 - \sigma(\boldsymbol{c}^\top \boldsymbol{x}_n))\right]^{1-z_n}$$

and:

$$p(y_n \mid \boldsymbol{x}_n, z_n; \boldsymbol{\theta}) = \left[\mathcal{N}(y_n \mid \boldsymbol{w}_1^\top \boldsymbol{x}_n, \sigma_1^2)\right]^{z_n} \left[\mathcal{N}(y_n \mid \boldsymbol{w}_0^\top \boldsymbol{x}_n, \sigma_0^2)\right]^{1-z_n}$$

(c) In the E-step, we compute the posterior over latent variables given the current parameter values. Give an expression for the posterior probability $p(z_n = 1 \mid \boldsymbol{x}_n, y_n; \boldsymbol{\theta})$. Do not replace the sigmoid with its definition $\sigma(x) = \frac{1}{1+\exp(-x)}$ or substitute the pdf of the normal distribution.

(d) Let $r_n = p(z_n = 1 \mid \boldsymbol{x}_n, y_n; \boldsymbol{\theta}^{\text{old}})$. Recall the objective for the M-step is given by:

$$\underset{\boldsymbol{\theta}}{\arg\max} \quad \mathbb{E}_{p(\boldsymbol{z} \mid \boldsymbol{y}, \boldsymbol{X}; \boldsymbol{\theta}^{\text{old}})}[\log p(\boldsymbol{y}, \boldsymbol{z} \mid \boldsymbol{X}; \boldsymbol{\theta})]$$

Simplify this expectation, substituting r_n where appropriate. Do not replace the sigmoid with its definition $\sigma(x) = \frac{1}{1+\exp(-x)}$ or substitute the pdf of the normal distribution.

(e) In the M-step, we maximize the expected log-joint with respect to the model parameters. For this model, it is difficult to jointly maximize with respect to $\{\boldsymbol{w}_0, \boldsymbol{w}_1\}$ and $\{\sigma_0^2, \sigma_1^2\}$. Instead, we will first maximize with respect to $\{\boldsymbol{w}_0, \boldsymbol{w}_1\}$ and then update $\{\sigma_0^2, \sigma_1^2\}$ using the updated $\{\boldsymbol{w}_0, \boldsymbol{w}_1\}$.

Using the objective above, find the M-step update for $\boldsymbol{w}_1$. Here, you will need to replace $\mathcal{N}$ with the normal pdf.

Hint: You only need to consider the parts of the expected log-joint which contain $\boldsymbol{w}_1$.

By analogy, write down the M-step update for $\boldsymbol{w}_0$. You do not need to show your work for this update.

(f) Find the M-step update for σ_1^2. Again, you will need to replace $\mathcal{N}$ with the normal pdf.

Hint: Maximize with respect to σ_1^2 and not σ_1.

By analogy, write down the M-step update for σ_0^2. You do not need to show your work for this update.

(g) *[Bonus]* Consider the M-step update for $\boldsymbol{c}$. Rewrite the objective from above, dropping additive constants which do not depend on $\boldsymbol{c}$. You should recognize this as (almost) the maximum likelihood objective for a familiar statistical model. Which model is this, and what makes the M-step objective for $\boldsymbol{c}$ slightly different? Is there a closed form update for $\boldsymbol{c}$? If so, write this closed form update. If not, explain what could be done instead to update $\boldsymbol{c}$.

Solution:

(a) True.

(b) The joint log-probability is:

$$\begin{aligned}
\log p(\boldsymbol{t}, \boldsymbol{z} \mid \boldsymbol{X}; \boldsymbol{\theta}) &= \sum_{n=1}^{N} \log p(t_n, z_n \mid \boldsymbol{x}_n; \boldsymbol{\theta}) \\
&= \sum_{n=1}^{N} \log\{\sigma(\boldsymbol{c}^\top \boldsymbol{x}_n)^{z_n}(1-\sigma(\boldsymbol{c}^\top \boldsymbol{x}_n))^{1-z_n} \\
&\qquad \mathcal{N}(t_n|\boldsymbol{w}_1^\top \boldsymbol{x}_n, \sigma_1^2)^{z_n} \mathcal{N}(t_n|\boldsymbol{w}_0^\top \boldsymbol{x}_n, \sigma_0^2)^{1-z_n}\} \\
&= \sum_{n=1}^{N} z_n \log \sigma(\boldsymbol{c}^\top \boldsymbol{x}_n) + (1-z_n)\log(1-\sigma(\boldsymbol{c}^\top \boldsymbol{x}_n)) + \\
&\qquad z_n \log \mathcal{N}(t_n|\boldsymbol{w}_1^\top \boldsymbol{x}_n, \sigma_1^2) + (1-z_n)\log \mathcal{N}(t_n|\boldsymbol{w}_0^\top \boldsymbol{x}_n, \sigma_0^2)
\end{aligned}$$

(c) Using Bayes Rule, we have:

$$\begin{aligned}
p(z=1|\boldsymbol{x}, t; \boldsymbol{\theta}) &= \frac{p(t, z=1|\boldsymbol{x}; \boldsymbol{\theta})}{p(t|\boldsymbol{x}; \boldsymbol{\theta})} \\
&= \frac{p(z=1|\boldsymbol{x}; \boldsymbol{\theta}) p(t|z=1, \boldsymbol{x}; \boldsymbol{\theta})}{p(z=0|\boldsymbol{x}; \boldsymbol{\theta}) p(t|z=0, \boldsymbol{x}; \boldsymbol{\theta}) + p(z=1|\boldsymbol{x}; \boldsymbol{\theta}) p(t|z=1, \boldsymbol{x}; \boldsymbol{\theta})}
\end{aligned}$$

Substituting in the model definition:

$$p(z=1|\boldsymbol{x}, t; \boldsymbol{\theta}) = \frac{\sigma(\boldsymbol{c}^\top \boldsymbol{x})\mathcal{N}(t|\boldsymbol{w}_1^\top \boldsymbol{x}, \sigma_1^2)}{(1-\sigma(\boldsymbol{c}^\top \boldsymbol{x}))\mathcal{N}(t|\boldsymbol{w}_0^\top \boldsymbol{x}, \sigma_0^2) + \sigma(\boldsymbol{c}^\top \boldsymbol{x})\mathcal{N}(t|\boldsymbol{w}_1^\top \boldsymbol{x}, \sigma_1^2)}$$

(d) Since $\mathbb{E}_{p(z_n|\boldsymbol{x}_n, t_n; \boldsymbol{\theta}^{\text{old}})}[z_n] = p(z_n = 1|\boldsymbol{x}_n, t_n; \boldsymbol{\theta}^{\text{old}}) = r_n$, using linearity of expectation, all we need to do is substitute z_n with r_n in the expression from part (a):

$$\begin{aligned}\sum_{n=1}^{N} r_n \log \sigma(\boldsymbol{c}^\top \boldsymbol{x}_n) + (1 - r_n) \log(1 - \sigma(\boldsymbol{c}^\top \boldsymbol{x}_n)) \\ + r_n \log \mathcal{N}(t_n|\boldsymbol{w}_1^\top \boldsymbol{x}_n, \sigma_1^2) + (1 - r_n) \log \mathcal{N}(t_n|\boldsymbol{w}_0^\top \boldsymbol{x}_n, \sigma_0^2)\end{aligned}$$

(e) To find the M-step update, we need to maximize the expected joint log-probability with respect to σ_1^2. First, we notice this is equivalent to maximizing:

$$\sum_{n=1}^{N} r_n \log \mathcal{N}(t_n|\boldsymbol{w}_1^\top \boldsymbol{x}_n, \sigma_1^2) = \sum_{n=1}^{N} r_n \left(-\frac{1}{2}\log(2\pi) - \frac{1}{2}\log \sigma_1^2 - \frac{1}{2\sigma_1^2}(t_n - \boldsymbol{w}_1^\top \boldsymbol{x}_n)) \right)$$

We take the derivative of this expression with respect to σ_1^2 and set it to zero:

$$\begin{aligned}\sum_{n=1}^{N} r_n \left(-\frac{1}{2\sigma_1^2} + \frac{1}{2(\sigma_1^2)^2}(t_n - \boldsymbol{w}_1^\top \boldsymbol{x}_n)^2 \right) = 0 &\implies \sum_{n=1}^{N} r_n \left(\frac{1}{2\sigma_1^2} \right) = \sum_{n=1}^{N} r_n \left(\frac{1}{2(\sigma_1^2)^2}(t_n - \boldsymbol{w}_1^\top \boldsymbol{x}_n)^2 \right) \\ &\implies \sum_{n=1}^{N} r_n = \sum_{n=1}^{N} r_n \left(\frac{1}{\sigma_1^2}(t_n - \boldsymbol{w}_1^\top \boldsymbol{x}_n)^2 \right) \\ &\implies \sigma_1^2 = \frac{\sum_{n=1}^{N} r_n (t_n - \boldsymbol{w}_1^\top \boldsymbol{x}_n)^2}{\sum_{n=1}^{N} r_n}\end{aligned}$$

By analogy, we find that the M-step update for σ_0^2 is:

$$\sigma_0^2 = \frac{\sum_{n=1}^{N} (1 - r_n)(t_n - \boldsymbol{w}_0^\top \boldsymbol{x}_n)^2}{\sum_{n=1}^{N} 1 - r_n}$$

(f) First, we drop additive constants with respect to $\boldsymbol{w}_1$ from the objective to find the update is given by:

$$\begin{aligned}\boldsymbol{w}_1 \leftarrow \operatorname*{argmax}_{\boldsymbol{w}_1} \quad & \sum_{n=1}^{N} r_n \log \mathcal{N}(t_n \mid \boldsymbol{w}_1^\top \boldsymbol{x}_n, \sigma_1^2) \\ = \operatorname*{argmax}_{\boldsymbol{w}_1} \quad & \sum_{n=1}^{N} r_n \left(-\frac{1}{2\sigma_1^2}(t_n - \boldsymbol{w}_1^\top \boldsymbol{x}_n)^2 \right) \\ = \operatorname*{argmin}_{\boldsymbol{w}_1} \quad & \sum_{n=1}^{N} r_n (t_n - \boldsymbol{w}_1^\top \boldsymbol{x}_n)^2\end{aligned}$$

This is a weighted least squares objective. We can rewrite it using matrix-vector notation:

$$\begin{aligned}\operatorname*{argmin}_{\boldsymbol{w}_1} \quad \sum_{n=1}^{N} r_n (t_n - \boldsymbol{w}_1^\top \boldsymbol{x}_n)^2 &= \operatorname*{argmin}_{\boldsymbol{w}_1} \quad (\boldsymbol{t} - \boldsymbol{X}\boldsymbol{w}_1)^\top \operatorname{diag}(\boldsymbol{r})(\boldsymbol{t} - \boldsymbol{X}\boldsymbol{w}_1) \\ &= \operatorname*{argmin}_{\boldsymbol{w}_1} \quad \boldsymbol{w}_1^\top \boldsymbol{X}^\top \operatorname{diag}(\boldsymbol{r})\boldsymbol{X}\boldsymbol{w}_1 - 2\boldsymbol{t}^\top \operatorname{diag}(\boldsymbol{r})\boldsymbol{X}\boldsymbol{w}_1\end{aligned}$$

Here, $\boldsymbol{r} = \begin{pmatrix} r_1 & \dots & r_n \end{pmatrix}^\top$. Taking the derivative with respect to $\boldsymbol{w}_1$ and setting to zero:

$$2X^\top \mathrm{diag}(\boldsymbol{r}) X \boldsymbol{w}_1 - 2X^\top \mathrm{diag}(\boldsymbol{r}) \boldsymbol{t} = 0 \implies \boldsymbol{w}_1 = (X^\top \mathrm{diag}(\boldsymbol{r}) X)^{-1} X^\top \mathrm{diag}(\boldsymbol{r}) \boldsymbol{t}$$

By analogy, the update for $\boldsymbol{w}_0$ is:

$$\boldsymbol{w}_0 = (X^\top \mathrm{diag}(\boldsymbol{1} - \boldsymbol{r}) X)^{-1} X^\top \mathrm{diag}(\boldsymbol{1} - \boldsymbol{r}) \boldsymbol{t}$$

(g) As a function of $\boldsymbol{c}$, the expected log probability is,

$$\begin{aligned}
\mathscr{L}(\boldsymbol{c}) &= \mathbb{E}_{p(\boldsymbol{z} \mid \boldsymbol{y}, X; \boldsymbol{\theta}^{\mathrm{old}}} \left[\sum_{n=1}^{N} \log p(z_n \mid \boldsymbol{x}_n; \boldsymbol{\theta}) \right] \\
&= \sum_{n=1}^{N} \mathbb{E}_{p(z_n \mid y_n, \boldsymbol{x}_n; \boldsymbol{\theta}^{\mathrm{old}})} \left[z_n \log \sigma(\boldsymbol{c}^\top \boldsymbol{x}_n) + (1 - z_n) \log \sigma(\boldsymbol{c}^\top \boldsymbol{x}_n) \right] \\
&= \sum_{n=1}^{N} \omega_n \boldsymbol{c}^\top \boldsymbol{x}_n - \log \left(1 + e^{\boldsymbol{c}^\top \boldsymbol{x}_n} \right).
\end{aligned}$$

where we have defined $\omega_n \triangleq p(z_n = 1 \mid \boldsymbol{x}_n, y_n; \boldsymbol{\theta})$. We recognize this as the log likelihood for logistic regression where the targets have been replaced with *expected* targets, ω_n.

Problem 2: *Variational Autoencoders*

Consider a VAE with non-negative latent variables,

$$\begin{aligned} z_{n,h} &\sim \mathrm{Exp}(\lambda) && \text{for } h = 1, \ldots, H \text{ and } n = 1, \ldots, N \\ \boldsymbol{x}_n &\sim \mathcal{N}(g(\boldsymbol{z}_n; \theta), \sigma^2 I) && \text{for } n = 1, \ldots, N \end{aligned}$$

where $\boldsymbol{x}_n \in \mathbb{R}^D$ are the observed data points, $\boldsymbol{z}_n = (z_{n,1}, \ldots, z_{n,H})^\top \in \mathbb{R}_+^H$ is the latent "encoding" of $\boldsymbol{x}_n$, and $g : \mathbb{R}_+^H \mapsto \mathbb{R}^D$ is the decoder with parameters θ. Assume a fixed form for the approximate variational posteriors,

$$q(\boldsymbol{z}_n; \boldsymbol{x}_n, \phi) = \prod_{h=1}^{H} \mathrm{Ga}(z_{n,h} \mid 2, f_h(\boldsymbol{x}_n; \phi)),$$

where $f_h : \mathbb{R}^D \mapsto \mathbb{R}_+$ is an encoder with parameters ϕ that outputs the rate parameter for the gamma posterior on the h-th coordinate of the encoding. Note that we are assuming the shape parameter of the gamma posterior is fixed to 2.

To learn the parameters of the parameters of the encoder (ϕ) and decoder (θ), we maximize the ELBO,

$$\mathscr{L}(\theta, \phi) = \sum_{n=1}^{N} \mathbb{E}_{q(\boldsymbol{z}_n; \boldsymbol{x}_n, \phi)} \left[\log p(\boldsymbol{x}_n \mid \boldsymbol{z}_n; \theta, \sigma^2) \right] - D_{\mathrm{KL}} \left(q(\boldsymbol{z}_n; \boldsymbol{x}_n, \phi) \,\|\, p(\boldsymbol{z}_n; \lambda) \right),$$

where $p(\boldsymbol{z}_n; \lambda)$ denotes the independent exponential prior from above.

(a) (1.5 pts) Is the following statement true or false? Explain your answer.

$$\nabla_\theta \mathscr{L}(\theta, \phi) = \sum_{n=1}^{N} \mathbb{E}_{q(\boldsymbol{z}_n; \boldsymbol{x}_n, \phi)} \left[\nabla_\theta \log p(\boldsymbol{x}_n \mid \boldsymbol{z}_n; \theta, \sigma^2) \right].$$

(b) (1.5 pts) Is the following statement true or false? Explain your answer.

$$\nabla_\phi \mathscr{L}(\theta, \phi) = \sum_{n=1}^{N} \mathbb{E}_{q(\boldsymbol{z}_n; \boldsymbol{x}_n, \phi)} \left[\nabla_\phi \log p(\boldsymbol{x}_n \mid \boldsymbol{z}_n; \theta, \sigma^2) \right] - \nabla_\phi D_{\mathrm{KL}} \left(q(\boldsymbol{z}_n; \boldsymbol{x}_n, \phi) \,\|\, p(\boldsymbol{z}_n; \lambda) \right).$$

(c) (5 pts) Derive a reparameterization trick for the gamma posterior. Specifically, write a sample $z_{n,h} \sim \mathrm{Ga}(2, f_h(\boldsymbol{x}_n; \phi))$ as a transformation of two uniform random variables, $u_1, u_2 \overset{\text{iid}}{\sim} \mathrm{Unif}([0, 1])$.

(d) (5 pts) Compute a closed form expression for the KL divergence in the ELBO.

$$D_{\mathrm{KL}} \left(q(\boldsymbol{z}_n; \boldsymbol{x}_n, \phi) \,\|\, p(\boldsymbol{z}_n; \lambda) \right) = \mathbb{E}_{q(\boldsymbol{z}_n; \boldsymbol{x}_n, \phi)} \left[\log q(\boldsymbol{z}_n; \boldsymbol{x}_n, \phi) - \log p(\boldsymbol{z}_n; \lambda) \right]$$

(e) (2 pts) In 1-2 sentences, explain how you would use the results from (c) and (d) to obtain an unbiased estimator of $\nabla_\phi \mathscr{L}(\theta, \phi)$.

Solution:

(a) TRUE. The second term has no θ dependence, and we can push the ∇_θ inside the first expectation because the measure that we are integrating wrt has no θ dependence.

(b) FALSE. The measure we are integrating wrt has ϕ dependence, so we have to be more careful about pushing ∇_ϕ inside the expectation, because are ignoring the effect it has on the measure.

(c)

$$\boxed{-\frac{\log u_1 + \log u_2}{f_h(\boldsymbol{x}_n;\phi)}.}$$

(d)

$$\begin{aligned}
\mathbb{E}_{q(\boldsymbol{z}_n;\boldsymbol{x}_n,\phi)}[\log q(\boldsymbol{z}_n;\boldsymbol{x}_n,\phi) - \log p(\boldsymbol{z}_n;\lambda)] &= \sum_{h=1}^{H} 2\log f_h(\boldsymbol{x}_n;\phi) - \log\Gamma(2) + \mathbb{E}_{z_{n,h}\sim q}[\log z_{n,h}] \\
&- f_h(\boldsymbol{x}_n;\phi)\cdot \mathbb{E}_{z_{n,h}\sim q}[z_{n,h}] - \log\lambda + \lambda\mathbb{E}_{z_{n,h}\sim q}[z_{n,h}] \\
&= \sum_{h=1}^{H} 2\log f_h(\boldsymbol{x}_n;\phi) - \log\Gamma(2) + \psi(2) - \log(f_h(\boldsymbol{x}_n,\theta)) \\
&- f_h(\boldsymbol{x}_n;\phi)\cdot\frac{2}{f_h(\boldsymbol{x}_n,\theta)} - \log\lambda + \lambda\frac{2}{f_h(\boldsymbol{x}_n,\theta)} \\
&= \boxed{\sum_{h=1}^{H} \log f_h(\boldsymbol{x}_n;\phi) - \log\lambda\Gamma(2) + \psi(2) - 2 + \frac{2\lambda}{f_h(\boldsymbol{x}_n;\phi)}.}
\end{aligned}$$

(e) In part d, we have just found a closed form for the KL divergence term, so we can just take its derivative wrt ϕ using autodiff. As for the expectation term, we can reparameterize the expectation to be over u_1 and u_2; pass the gradient wrt ϕ inside of this expectation; and then simply sample u_1 and u_2 to get an unbiased estimator for this derivative.

Problem 3: *Hidden Markov Models*

Consider a hidden Markov model (HMM),

$$\begin{aligned} z_1 &\sim \mathrm{Cat}(\pi_0) & \\ z_t &\sim \mathrm{Cat}(\pi_{z_{t-1}}) & \text{for } t = 2, \dots, T \\ \boldsymbol{x}_t &\sim p(\boldsymbol{x} \mid \boldsymbol{\theta}_{z_t}) & \text{for } t = 1, \dots, T \end{aligned}$$

where $z_t \in \{1, \dots, K\}$ denotes the discrete latent state at time t. The HMM parameters consist of the initial state probabilities, $\boldsymbol{\pi}_0$, the transition matrix $\boldsymbol{\Pi}$ with rows $\boldsymbol{\pi}_k \in \Delta_K$ for $k = 1, \dots, K$, and the emission parameters $\{\boldsymbol{\theta}_k\}_{k=1}^K$.

We can represent the latent states in terms of their states at the times of change points, $c_n \in \{1, \dots, K\}$, and the corresponding durations between change points, $d_n \in \mathbb{N}_+$ (these are positive integers). Let $c_1 = z_1$ and let d_1 denote the number of time steps before the state changes. Then let $c_2 = z_{1+d_1}$ and let d_2 be the number of time steps before the state changes again. For example, this state sequence would be represented as follows,

$$\boldsymbol{z}_{1:T} = \underbrace{3,3,3,3,3}_{c_1=3, d_1=5}, \underbrace{1,1,1,1}_{c_2=1, d_2=4}, \underbrace{2,2,2,2,2,2}_{c_3=2, d_3=6}, \underbrace{1,1,1,1,1}_{c_4=1, d_4=5}.$$

(a) (5 pts) Derive the probability mass function $p(d_n \mid c_n = k)$.

(b) (5 pts) Derive the probability mass function $p(c_{n+1} \mid c_n = k)$.

(c) (5 pts) Suppose we want to model a system in which the emissions are conditionally independent given the latent state, as above, but where durations are uniformly distributed between 1 and 3 time steps. That is, for all k,

$$p(d_n = i \mid c_n = k) = \begin{cases} \frac{1}{3} & \text{if } i = 1, 2, 3 \\ 0 & \text{otherwise} \end{cases}$$

How could model such a system using an HMM on an extended state space? Specifically, let $z'_t \in \{1, \dots, K'\}$ denote the states of your HMM; you may have $K' > K$. What are the transition and emission probabilities?

(d) ( ) What is the computational complexity of the forward-backward algorithm for this HMM?

Solution:

(a) The duration is following a geometric distribution, so

$$\boxed{p(d_n | c_n = k) = (1 - \pi_{k,k}) \cdot \pi_{k,k}^{d_n - 1}.}$$

(b) The idea behind a change point is that $c_{n+1} \neq c_n$. So, if $c_n = k$, then c_{n+1} is a categorical distribution on all the states *except* k. We can write this distribution as

$$\boxed{\mathbb{P}(c_{n+1} = i) = \frac{\pi_{ki} \mathbb{1}(i \neq k)}{1 - \pi_{k,k}}.}$$

(c) The extended state space now takes the form of the Cartesian product $\{1, \ldots, K\} \times \{1, 2, 3\}$, where we refer to the first coordinate as z and the second coordinate as i.

The emission probabilities are entirely governed by z; that is $x_t \sim p(\boldsymbol{x}|\boldsymbol{\theta}_{z_t})$.

The transition probabilities are as follows.

If $i = 1$, w.p. $2/3$ we transition to the state $(z_t, 2)$, and w.p. $1/3$ we transition to a state $(z_{t+1}, 1)$, where $z_{t+1} \neq z_t$, and the transition probabilities given by $\mathbb{P}(z_{t+1} = k) = \frac{\pi_k}{\sum_{j \neq z_t} \pi_j}$.

If $i = 2$, then w.p. $1/2$ we transition to the state $(z_t, 3)$, and w.p. $1/2$ we transition to a state $(k, 1)$, where $k \neq z_t$. The probabilities of transitioning to these different values of k are governed by $\mathbb{P}(z_{t+1} = k) = \frac{\pi_k}{\sum_{j \neq z_t} \pi_j}$.

OTOH, if $i = 3$, then we transition to a state with $i = 1$, and $z_{t+1} \neq z_t$, and with probabilities given by $\mathbb{P}(z_{t+1} = k) = \frac{\pi_k}{\sum_{j \neq z_t} \pi_j}$.

(d) The complexity of the forward-backward algorithm for a standard HMM is $O(TK^2)$. Our embedding of this hidden semi-Markov model has $3K$ states, so it's the same order of magnitude.

More generally, a semi-Markov model with a uniform distribution of durations $d_n \sim \text{Unif}(1, \ldots, R)$, would have complexity $O(TK^2R^2)$.

Problem 4: *Recurrent Neural Networks*

Consider a stochastic RNN with linear hidden state dynamics

$$h_{t+1} \sim \mathrm{N}(ah_t, 1),$$

and initial distribution $h_0 \sim \mathrm{N}(0, 1)$. This is called a Gaussian linear dynamical system (LDS). It's also similar to the noising process we studied in the context of denoising diffusion models.

(a) Derive the conditional distribution, $p(h_T \mid h_0)$?

(b) Suppose we receive one observation, $y_T \sim \mathrm{N}(h_T, 1)$. Compute the posterior, $p(h_0 \mid y_T)$.

(c) In this case the posterior has a closed form, but suppose we didn't know that and we tried to solve for the posterior mode $h_0^{(\mathsf{MAP})}$ using gradient ascent. Compute the gradient $\nabla_{h_0} \log p(h_0, y_T)$.

(d) For $a < 1$, how does the gradient scale with T, and why could that be problematic for gradient based learning?

Solution:

(a) The forward conditionals can be computed recursively. Assume $p(h_t \mid h_0) = \mathrm{N}(\lambda_t h_0, \sigma_t^2)$. Then,

$$\begin{aligned} p(h_{t+1} \mid h_0) &= \int p(h_{t+1} \mid h_t)\, p(h_t \mid h_0)\, \mathrm{d}h_T \\ &= \int \mathrm{N}(h_{t+1} \mid ah_t, 1)\, \mathrm{N}(h_t \mid \lambda_t h_0, \sigma_t^2)\, \mathrm{d}h_T \\ &= \mathrm{N}(a\lambda_t h_0, a^2\sigma_t^2 + 1), \end{aligned}$$

which shows that

$$\begin{aligned} \lambda_{t+1} &= a\lambda_t, \\ \sigma_{t+1}^2 &= a^2\sigma_t^2 + 1. \end{aligned}$$

With the base case $\lambda_1 = a$, we can unwind the recursion to obtain,

$$\lambda_t = a^t$$

Likewise, with $\sigma_1^2 = 1$ we find,

$$\begin{aligned} \sigma_2^2 &= a^2 + 1 \\ \sigma_3^2 &= a^4 + a^2 + 1 \\ \sigma_4^2 &= a^6 + a^4 + a^2 + 1 \\ \sigma_t^2 &= \sum_{i=0}^{t-1} a^{2i} = \frac{1 - a^{2t}}{1 - a^2}. \end{aligned}$$

Note that for $a < 1$, the conditional variance converges to a finite value. For $a > 1$, it diverges.

(b) First apply the sum rule of probability to obtain the likelihood,

$$\begin{aligned} p(y_T \mid h_0) &= \int p(y_T, h_T \mid h_0) \, \mathrm{d}h_T \\ &= \int \mathrm{N}(y_T \mid h_T, 1) \, \mathrm{N}(h_T \mid \lambda_T h_0, \sigma_T^2) \, \mathrm{d}h_T \\ &= \mathrm{N}(y_T \mid \lambda_T h_0, \sigma_T^2 + 1), \end{aligned}$$

in the notation of our solution to (a).

Now apply Bayes' rule,

$$\begin{aligned} p(h_0 \mid y_T) &\propto p(h_0) p(y_T \mid h_0) \\ &\propto \mathrm{N}(h_0 \mid 0, 1) \, \mathrm{N}(y_T \mid \lambda_T h_0, \sigma_T^2 + 1) \\ &\propto \exp\left\{ -\frac{1}{2} h_0^2 - \frac{1}{2(\sigma_T^2 + 1)} (y_T - \lambda_T h_0)^2 \right\} \\ &\propto \exp\left\{ -\frac{h_0^2}{2} \left(1 + \frac{\lambda_T^2}{\sigma_T^2 + 1} \right) + h_0 \left(\frac{\lambda_T y_T}{\sigma_T^2 + 1} \right) \right\} \end{aligned}$$

Completing the square yields,

$$\begin{aligned} p(h_0 \mid y_T) &= \mathrm{N}(\tilde{\mu}, \tilde{\sigma}^2) \\ \tilde{\sigma}^2 &= \left(1 + \frac{\lambda_T^2}{\sigma_T^2 + 1} \right)^{-1} \\ \tilde{\mu} &= \tilde{\sigma}^2 \left(\frac{\lambda_T y_T}{\sigma_T^2 + 1} \right). \end{aligned}$$

(c) The log joint probability is equal to the log posterior plus a constant with respect to h_0,

$$\log p(h_0, y_T) = -\frac{1}{2\tilde{\sigma}^2} (h_0 - \tilde{\mu})^2 + c.$$

Taking the derivative with respect to h_0 yields,

$$\frac{\mathrm{d}}{\mathrm{d}h_0} \log p(h_0, y_T) = \frac{\tilde{\mu} - h_0}{\tilde{\sigma}^2}$$

(d) The gradient of the log joint is the gradient of the prior plus the gradient of the likelihood. The gradient of the prior is simply $\nabla \log p(h_0) = -h_0$. The gradient of the likelihood is,

$$\nabla_{h_0} \log \mathrm{N}(y_T \mid \lambda_T h_0, \sigma_T^2 + 1) = \left(\frac{y_T - \lambda_T h_0}{\sigma_T^2 + 1} \right) \lambda_T.$$

Consider the case where $a < 1$. Then $\lim_{T \to \infty} \lambda_T = \lim_{T \to \infty} a^T = 0$, whereas $\lim_{T \to \infty} \sigma_T^2 = (1 - a^2)^{-1}$ is a positive constant. Thus, the gradient of the likelihood goes to zero.

In this limit, with $a < 1$, we see that the posterior reduces to the prior, and the information about y_T conveyed by the likelihood gradient vanishes.