STATS 305B: Final Exam

Write your name here:

Instructions:

- Your grade will come from your **best 5 out of 7 questions.** Each question is worth 20 points so the max score is 100.
- 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. Most questions have a relatively simple answer. If you find yourself doing integration by parts, rethink your approach.

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: *Exponential Family Distributions (20 pts)*

Consider the negative binomial distribution for a random variable $X \in \mathbb{N}$ parameterized by $r \in \mathbb{R}_+$ and $\rho \in [0,1]$. Its pmf is,

$$\mathrm{NB}(x; r, \rho) = \frac{\Gamma(r+x)}{x!\Gamma(r)} \rho^x (1-\rho)^r,$$

where $\Gamma(\cdot)$ is the gamma function.

(a) (4 pts) Assume r is fixed. Write the negative binomial pmf in exponential family form. What is its natural parameter η, sufficient statistic $t(x)$, and log normalizer $A(\eta)$?

(b) (4 pts) Using the log normalizer, compute the expected value $\mathbb{E}[X]$ where $X \sim \mathrm{NB}(r, \rho)$.

(c) (4 pts) Using the log normalizer, compute the variance $\mathrm{Var}[X]$ where $X \sim \mathrm{NB}(r, \rho)$.

(d) (4 pts) Suppose $\lambda \sim \mathrm{Ga}(r, \frac{1-\rho}{\rho})$ and $X \mid \lambda \sim \mathrm{Po}(\lambda)$, where Ga denotes the gamma distribution with density $\mathrm{Ga}(\lambda; a, b) = \frac{b^a}{\Gamma(a)} \lambda^{a-1} e^{-b\lambda}$, and where Po denotes the Poisson distribution with pmf $\mathrm{Po}(x; \lambda) = \frac{1}{x!} e^{-\lambda} \lambda^x$. Compute the pmf of the marginal distribution,

$$p(x; r, \rho) = \int \mathrm{Po}(x; \lambda)\, \mathrm{Ga}(\lambda; r, \tfrac{1-\rho}{\rho})\, \mathrm{d}\lambda$$

(e) (4 pts) The ratio of the variance to the mean is a common index of a distribution's *dispersion*. Consider a negative binomial distribution and a Poisson distribution with the same mean. Which has higher dispersion? Explain why your answer makes sense in light of Problem 1d.

Solution:

(a)

$$\mathrm{NB}(x; r, \rho) = \frac{\Gamma(r+x)}{\Gamma(x+1)\Gamma(r)} \cdot \exp(\log(\rho) \cdot x + r \log(1-\rho))$$

So, we see that

$$\begin{aligned} \eta &= \log(\rho) \\ t(x) &= x \\ A(\eta) &= -r \log(1 - \exp(\eta)). \end{aligned}$$

(b)

$$\mathbb{E}[X] = r \frac{\rho}{1-\rho}$$

(c)

$$\mathrm{Var}[X] = r \frac{\rho}{(1-\rho)^2}$$

(d)

$$\begin{aligned}
p(x;r,\rho) &= \int \frac{1}{\Gamma(x+1)}\exp(-\lambda)\lambda^x \frac{((1-\rho)/\rho)^r}{\Gamma(r)}\lambda^{r-1}\exp(-(1-\rho)\lambda/\rho) \\
&= \frac{1}{\Gamma(x+1)\Gamma(r)}\cdot(1-\rho)^r\cdot\frac{1}{\rho^r}\int \lambda^{x+r-1}\exp-(1/\rho)\lambda \\
&= \frac{1}{\Gamma(x+1)\Gamma(r)}\cdot(1-\rho)^r\cdot\frac{1}{\rho^r}\cdot\Gamma(x+r)\cdot\rho^{x+r} \\
&= \frac{\Gamma(x+r)}{\Gamma(x+1)\Gamma(r)}(1-\rho)^r\rho^x. \\
&= \boxed{\mathrm{NB}(x;r,\rho).}
\end{aligned}$$

(e) Negative binomial has higher dispersions. Makes sense bc problem 1d shows that NBin is mixture of many different Pois w different rates, and some of those rates might be quite large, which will make NB overdispersed relative to Pois.

Problem 2: *Logistic Regression (20 pts)*

Consider a logistic regression model,

$$p(y_i \mid \boldsymbol{x}_i) = \mathrm{Bern}(y_i \mid \sigma(\boldsymbol{\beta}^\top \boldsymbol{x}_i))$$

where $\boldsymbol{\beta} \in \mathbb{R}^2$ and $\boldsymbol{x}_i \in \mathbb{R}^2$ are two-dimensional weights and covariates, respectively, and where $\sigma(a) = 1/(1+e^{-a})$ is the logistic function. Suppose you observe the following set of independent observations,

$x_{i,1}$	$x_{i,2}$	y_i
1	0	0
0	1	1
-1	0	0
0	-1	1

Table 1: Four data points for Problem 2

(a) (1 pt) Sketch the data with an "x" where $y_i = 1$ and an "o" where $y_i = 0$.

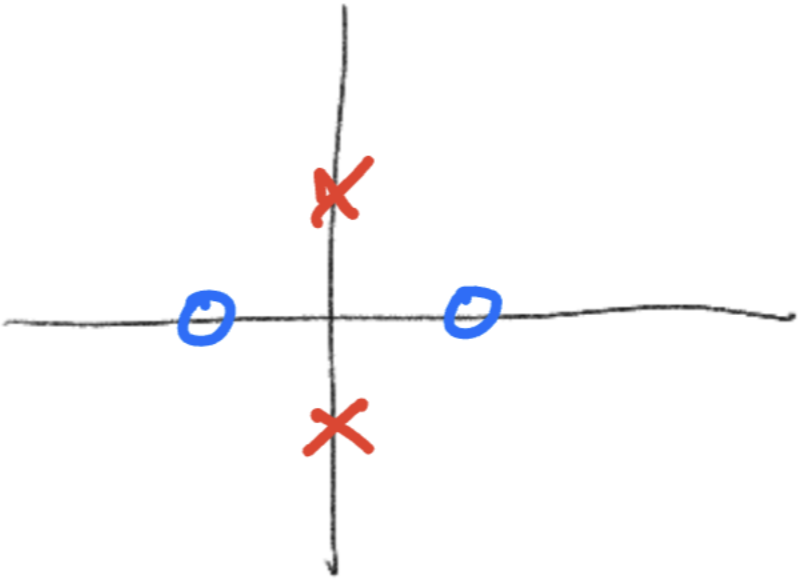
(b) (5 pts) Write the log likelihood function, $\mathscr{L}(\boldsymbol{\beta})$, and simplify.

(c) (5 pts) Sketch the function $\log(1+e^a) + \log(1+e^{-a})$ for $a \in \mathbb{R}$. Label your axes and key features of the graph, like the y-intercept.

(d) (5 pts) Sketch the contours of the log likelihood $\mathscr{L}(\boldsymbol{\beta})$ for $\boldsymbol{\beta} \in \mathbb{R}^2$.

(e) (4 pts) Solve for the MLE $\hat{\boldsymbol{\beta}}_{\mathsf{MLE}}$.

Solution:

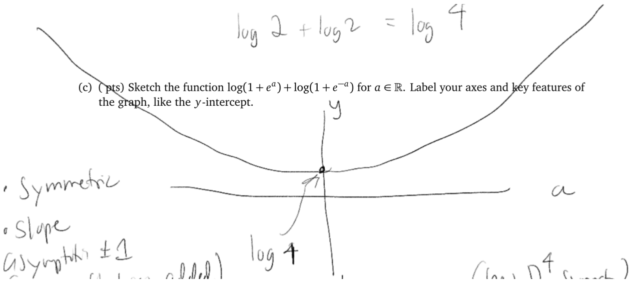


(a)

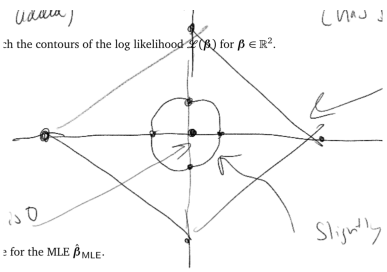
(b)

$$
\begin{aligned}
\mathscr{L}(\boldsymbol{\beta}) &= \log((1-\sigma(\beta_1))) + \log((1-\sigma(-\beta_1))) + \log(\sigma(\beta_2)) + \log((\sigma(-\beta_2))) \\
&= \log(\sigma(-\beta_1)) + \log(\sigma(\beta_1)) + \log(\sigma(\beta_2)) + \log(\sigma(-\beta_2)) \\
&= \boxed{-\Big(\log(1+\exp\beta_1) + \log(1+\exp-\beta_1) + \log(1+\exp\beta_2) + \log(1+\exp-\beta_2)\Big).}
\end{aligned}
$$

(c)


(c) (pts) Sketch the function $\log(1+e^{a})+\log(1+e^{-a})$ for $a\in\mathbb{R}$. Label your axes and key features of the graph, like the y-intercept.

(d)


ch the contours of the log likelihood $\mathscr{L}(\boldsymbol{\beta})$ for $\boldsymbol{\beta}\in\mathbb{R}^2$.

e for the MLE $\hat{\boldsymbol{\beta}}_{\mathrm{MLE}}$.

(e) $(0,0)$

Problem 3: *Mixture Models (20 pts)*

Let $\boldsymbol{y}_i \in \mathbb{N}^D$ be a vector of counts. For example, it could correspond to the number of spikes measured in each of D neurons in the i-th time bin. Let $z_i \in \{1, \dots, K\}$ denote a discrete class assignment for that corresponding measurement. Consider the following mixture model,

$$p(\{\boldsymbol{y}_i, z_i\}_{i=1}^n \mid \boldsymbol{\pi}, \{\boldsymbol{\lambda}_k\}_{k=1}^K) = \prod_{i=1}^n \mathrm{Cat}(z_i \mid \boldsymbol{\pi}) \prod_{d=1}^D \mathrm{Po}(y_{i,d} \mid \lambda_{z_i,d})$$

where $\boldsymbol{\pi} \in \Delta_{K-1}$ is a distribution over the K classes and $\boldsymbol{\lambda}_k = (\lambda_{k,1}, \dots, \lambda_{k,D})^\top$ is a vector of rates corresponding to class k. The goal is to obtain a maximum likelihood estimate of $\{\boldsymbol{\lambda}_k\}_{k=1}^K$, assuming $\boldsymbol{\pi}$ is known.

(a) (10 pts) **E step:** Compute the *responsibilities*, $\omega_{i,k} = \Pr(z_i = k \mid \boldsymbol{y}_i, \boldsymbol{\pi}, \{\boldsymbol{\lambda}_j\}_{j=1}^K)$.

(b) (10 pts) **M step:** solve for the parameters $\boldsymbol{\lambda}_k$ that maximize the expected log likelihood, in terms of the responsibilities from part (a).

Solution:

1. By Bayes' rule

$$\Pr(z_i = k | y_k, \pi, \vec{\lambda}) \propto \pi_k \prod_{d=1}^D \mathrm{Po}(y_{i,d} | \lambda_{k,d}).$$

So, it follows that

$$\boxed{\omega_{i,k} = \frac{\pi_k \prod_{d=1}^D \mathrm{Po}(y_{i,d}|\lambda_{k,d})}{\sum_{k=1}^K \pi_k \prod_{d=1}^D \mathrm{Po}(y_{i,d}|\lambda_{k,d})}.}$$

2. We are told to find $\vec{\lambda}$ that maximizes

$$\begin{aligned}\mathbb{E}\Big[\log \prod_{i=1}^n \prod_{d=1}^D \mathrm{Po}(y_{i,d}|\lambda_{z_i,d})\Big] &= \mathbb{E}\Big[\sum_{i,d} \log \mathrm{Po}(y_{i,d}|\lambda_{z_i,d})\Big] \\ &= \sum_{i,d,k} \mathbb{E}[\mathbb{1}(z_i = k)] \log \mathrm{Po}(y_{i,d}|\lambda_{k,d}) \\ &= \sum_{i,d,k} \omega_{i,k} \log \mathrm{Po}(y_{i,d}|\lambda_{k,d}) \\ &\doteq \sum_{i,d,k} \omega_{i,k}(-\lambda_{k,d} + y_{i,d} \log \lambda_{k,d})\end{aligned}$$

Taking the derivative of this quantity wrt $\lambda_{k,d}$ gives

$$\sum_i \omega_{i,k} = \sum_i \frac{\omega_{i,k} y_{i,d}}{\lambda^*_{k,d}}$$

or

$$\boxed{\lambda^*_{k,d} = \frac{\sum_i \omega_{i,k} y_{i,d}}{\sum_i \omega_{i,k}}.}$$

Problem 4: *Variational Autoencoders (20 pts)*

Consider the following generative model for observations $\boldsymbol{x} \in \mathbb{R}^D$ and latent variables $z \in \mathbb{R}$,

$$p(z, \boldsymbol{x}) = \mathrm{N}(z \mid 0, 1)\, \mathrm{N}(\boldsymbol{x} \mid \boldsymbol{\theta} \cdot z, \sigma^2 \boldsymbol{I}),$$

where the $\boldsymbol{\theta} \in \mathbb{R}^D$ is a parameter to be learned, and σ^2 is a fixed hyperparameter. Assume the amortized posterior (aka encoder) is of the form,

$$q(z \mid \boldsymbol{x}) = \mathrm{N}(z \mid \boldsymbol{\phi}^\top \boldsymbol{x}, \nu^2),$$

where $\boldsymbol{\phi} \in \mathbb{R}^D$ and $\nu^2 \in \mathbb{R}_+$ are variational parameters to be learned. This is a variational autoencoder (VAE) with a **linear** encoder and decoder.

(a) (7 pts) Write the local ELBO $\mathscr{L}(\boldsymbol{\theta}, \boldsymbol{\phi}, \nu) \leq \log p(\boldsymbol{x}; \boldsymbol{\theta})$ for a single observation $\boldsymbol{x}$. Leave your answer in terms of the Gaussian density function.

(b) (2 pts) When you maximize the ELBO, what quantity is minimized?

(c) (9 pts) Fixing $\boldsymbol{\theta}$, solve for the variational parameters $\boldsymbol{\phi}^\star$ and $\nu^\star$ that maximize the ELBO.

(d) (2 pts) With the optimal variational parameters found above, is the ELBO tight? I.e., does $\mathscr{L}(\boldsymbol{\theta}, \boldsymbol{\phi}^\star, \nu^\star)$ equal $\log p(\boldsymbol{x}; \boldsymbol{\theta})$?

Solution:

(a)

$$\begin{aligned}
\ln p(\boldsymbol{x}; \boldsymbol{\theta}) &= \ln\left(\sum_z p(\boldsymbol{x}, z; \boldsymbol{\theta})\right) \\
&= \ln\left(\sum p(\boldsymbol{x}, z) \frac{q(z|\boldsymbol{x})}{q(z|\boldsymbol{x}}\right) \\
&= \ln \mathbb{E}_{q(z|\boldsymbol{x})}\left[\frac{p(\boldsymbol{x}, z)}{q(z|\boldsymbol{x})}\right] \\
&\geq \mathbb{E}_{q(z|\boldsymbol{x})}\left[\ln p(\boldsymbol{x}, z) - \ln q(z|\boldsymbol{x})\right] \\
&= \boxed{\mathbb{E}_{q(z|\boldsymbol{x})}\left[\mathrm{N}(z|0,1) + \ln \mathrm{N}(\boldsymbol{x}|\boldsymbol{\theta} \cdot z, \sigma^2 \boldsymbol{I}) - \ln \mathrm{N}(z|\boldsymbol{\phi}^\top \boldsymbol{x}, \nu^2)\right]}
\end{aligned}$$

(b)

$$\begin{aligned}
\ln p(\boldsymbol{x}; \boldsymbol{\theta}) &\geq \mathbb{E}_{q(z|\boldsymbol{x})}\left[\ln p(x) + \ln p(z|\boldsymbol{x}) - \ln q(z|\boldsymbol{x})\right] \\
&= \ln p(\boldsymbol{x}) + \mathbb{E}_{q(z|\boldsymbol{x})}\left[\ln p(z|\boldsymbol{x}) - \ln q(z|\boldsymbol{x})\right] \\
&= \ln p(x) - \mathrm{KL}(qz|\boldsymbol{x})||p(z|\boldsymbol{x}))
\end{aligned}$$

Maximizing the ELBO is equivalent to minimizing the KL divergence of the posterior $p(z|\boldsymbol{x})$ and variational distribution $q(z|\boldsymbol{x})$

(c) The prior and likelihood are both Normal so we can recognize that the posterior will also be normal. We maximize the ELBO by finding the posterior parameters.

$$\begin{aligned}
p(z|\boldsymbol{x}) &\propto p(z,\boldsymbol{x}) \\
&\propto \exp\left\{-\frac{1}{2}z^2 - \frac{1}{2\sigma^2}\|\boldsymbol{x}-\boldsymbol{\theta}z\|^2\right\} \\
&\propto \exp\left\{-\frac{1}{2}z^2\left(1+\frac{1}{\sigma^2}\boldsymbol{\theta}^\top\boldsymbol{\theta}\right)+z\left(\frac{\boldsymbol{\theta}^\top\boldsymbol{x}}{\sigma^2}\right)\right\} \\
&\propto \mathrm{N}\left(\left(\frac{\boldsymbol{\theta}}{\sigma^2+\boldsymbol{\theta}^\top\boldsymbol{\theta}}\right)^\top\boldsymbol{x}, \frac{\sigma^2}{\sigma^2+\boldsymbol{\theta}^\top\boldsymbol{\theta}}\right)
\end{aligned}$$

$$\boxed{\boldsymbol{\phi}^* = \frac{\boldsymbol{\theta}}{\sigma^2+\boldsymbol{\theta}^\top\boldsymbol{\theta}}, \qquad \nu^* = \frac{\sigma}{\sqrt{\sigma^2+\boldsymbol{\theta}^\top\boldsymbol{\theta}}}}$$

(d) The ELBO is tight because the posterior and variational distribution are both Normal.

Problem 5: *Hidden Markov Models (HMMs) (20 pts)*

A *factorial* HMM is a hidden Markov model with $L \geq 2$ levels of latent states. The latent states evolve independently at each level, but they all jointly determine the likelihood for the observations. Let $\boldsymbol{z}^{(\ell)} = (z_1^{(\ell)}, \dots, z_T^{(\ell)})$ denote the sequence of latent states at level ℓ. Assume that for all levels and time steps, the discrete latent states take values $z_t^{(\ell)} \in \{1, \dots, K\}$. Let $\boldsymbol{x} = (x_1, \dots, x_T)$ denote the sequence of observations. The joint distribution factors as,

$$p(\boldsymbol{x}, \{\boldsymbol{z}^{(\ell)}\}_{\ell=1}^{L}) = \left[\prod_{\ell=1}^{L} p(z_1^{(\ell)}) \prod_{t=2}^{T} p(z_t^{(\ell)} \mid z_{t-1}^{(\ell)})\right] \times \prod_{t=1}^{T} p(x_t \mid z_t^{(1)}, \dots, z_t^{(L)}).$$

(a) (7 pts) Write the factorial HMM above as a standard HMM on an extended state space. (Define your extended state space and the corresponding transition probabilities.)

(b) (6 pts) What is the time complexity of the forward-backward for a factorial HMM? Your answer should be in terms of T, K, and L.

(c) (7 pts) Consider a factorial HMM with binary latent states, $z_t^{(\ell)} \in \{0, 1\}$. Let $\boldsymbol{z}_t = (z_t^{(1)}, \dots, z_t^{(L)}) \in \{0, 1\}^L$ denote the vector of latent states at time t. Assume a Gaussian likelihood,

$$p(x_t \mid z_t^{(1)}, \dots, z_t^{(L)}; \boldsymbol{\theta}) = \mathrm{N}\left(x_t \mid \boldsymbol{\theta}^\top \boldsymbol{z}_t, 1\right).$$

where $\boldsymbol{\theta} = (\theta_1, \dots, \theta_L) \in \mathbb{R}^L$. Intuitively, the mean of x_t is a sum of contributions θ_l for each level that is "on" at time t.

With a small number of levels (say, $L < 10$), we can use the forward-backward algorithm to compute posterior expectations $\boldsymbol{\omega}_t = \mathbb{E}[\boldsymbol{z}_t \mid \boldsymbol{x}]$ and $\boldsymbol{\Omega}_t = \mathbb{E}[\boldsymbol{z}_t \boldsymbol{z}_t^\top \mid \boldsymbol{x}]$ for all $t = 1, \dots, T$. Find a closed-form solution for the M-step,

$$\boldsymbol{\theta}^\star = \arg\max_{\boldsymbol{\theta}} \mathbb{E}_{p(\boldsymbol{z}|\boldsymbol{x})}\left[\sum_{t=1}^{T} \log p(x_t \mid z_t^{(1)}, \dots, z_t^{(L)}; \boldsymbol{\theta})\right].$$

Your answer should be in terms of $\boldsymbol{\omega}_t$ and/or $\boldsymbol{\Omega}_t$.

Solution:

(a)

$$p(\boldsymbol{x}, \{\boldsymbol{z}^{(\ell)}\}_{\ell=1}^{L}) = \prod_{t=1}^{T} p(x_t|\boldsymbol{z}_t^{(1)}, \dots, \boldsymbol{z}_t^{(L)}) p(\boldsymbol{z}_t^{(1)}, \dots, \boldsymbol{z}_t^{(L)})$$

$$= p(\boldsymbol{z}_1^{(1)}, \dots, \boldsymbol{z}_1^{(L)}) \prod_{t=2}^{T} p(\boldsymbol{z}_t^{(1)}, \dots, \boldsymbol{z}_t^{(L)}) \prod_{t=1}^{T} p(\boldsymbol{x}_t|\boldsymbol{z}_t^{(1)}, \dots, z_t^{(L)})$$

We can define $\tilde{\boldsymbol{z}}_t := (\boldsymbol{z}_t^{(1)}, \dots, \boldsymbol{z}_t^{(L)})$:

$$= p(\tilde{\boldsymbol{z}}_1) \prod_{t=2}^{T} p(\tilde{\boldsymbol{z}}_t|\boldsymbol{z}_{t-1}) \prod_{t=1}^{T} p(\boldsymbol{x}_t|\boldsymbol{z}_t)$$

Let P_ℓ be the $K \times K$ transition matrix for the Markov chain at level ℓ. The new transition matrix P will be $K^L \times K^L$ and can be written using a Kronecker product: $P = P_1 \otimes \cdots \otimes P_L$.

(b) We want to compute forward messages $\alpha_{t+1} = P^\top(\alpha_t \odot \ell_t)$

The dominant term at each time step is computing $P^\top v$ where v is some K^L dimensional vector. Let V be such that $v = \text{vec}(V)$. Then $(P_1 \otimes P_2)v = \text{vec}(P_2 V P_1^\top)$ and the complexity of this operation is $O(K^L)$. We can use this to write a recurrence for our time complexity. Let $T(L)$ be the time it takes to compute $(P_1 \otimes \cdots \otimes P_L)v$. Then $T(L) = KT(L-1) + K^{L+1}$. Solving this recurrence gives $T(L) = K(KT(L-2) + K^L) + K^{L+1} = K^2 T(L-2) + 2K^{L+1} = \ldots = O(LK^{L+1})$. Repeating for each time step gives $\boxed{O(TLK^{L+1})}$

Naive solution that only receives most points: Computing $\alpha_t \cdot \ell_t$ takes $O(K^L)$, $P^\top(\alpha_t \cdot \ell_t)$ takes $O(K^{2L})$. Iteratively computing forward messages for T observations costs a total of $O(TK^{2L})$

(c)

$$\begin{aligned}
\mathscr{L} &:= \mathbb{E}_{p(\boldsymbol{z}|\boldsymbol{x})}\left[\sum_{t=1}^{T} \log p(x_t \mid z_t^{(1)}, \ldots, z_t^{(L)}; \boldsymbol{\theta})\right] \\
&= \int \left(\frac{T}{2}\ln(2\pi) - \frac{1}{2}\sum_{t=1}^{T}(x_t - \boldsymbol{\theta}^\top \boldsymbol{z}_t)^2\right) p(\boldsymbol{z}|x)d\boldsymbol{z} \\
&= \frac{T}{2}\ln(2\pi) - \int \left(\frac{1}{2}\sum_{t=1}^{T}(x_t - \boldsymbol{\theta}^\top \boldsymbol{z}_t)^2\right) p(\boldsymbol{z}|x)d\boldsymbol{z}
\end{aligned}$$

$$\begin{aligned}
\nabla_{\boldsymbol{\theta}}\mathscr{L} &= -\frac{1}{2}\sum_{t=1}^{T}\int \nabla_{\boldsymbol{\theta}}\left[(x_t - \boldsymbol{\theta}^\top \boldsymbol{z}_t)^2 p(\boldsymbol{z}|x)\right] d\boldsymbol{z} \\
&= -\frac{1}{2}\sum_{t=1}^{T}\int \left(\nabla_{\boldsymbol{\theta}}\left[(x_t - \boldsymbol{\theta}^\top \boldsymbol{z}_t)^2\right]\right) p(\boldsymbol{z}|x)d\boldsymbol{z} - \frac{1}{2}\sum_{t=1}^{T}\int (x_t - \boldsymbol{\theta}^\top \boldsymbol{z}_t)^2 \nabla_{\boldsymbol{\theta}}\left[p(\boldsymbol{z}|x)\right] d\boldsymbol{z} \\
&= -\frac{1}{2}\sum_{t=1}^{T}\int 2(x_t - \boldsymbol{\theta}^\top \boldsymbol{z}_t)\boldsymbol{z}_t p(\boldsymbol{z}|x)d\boldsymbol{z} \\
&= -\sum_{t=1}^{T}\mathbb{E}_{p(\boldsymbol{z}|x)}\left[x_t \boldsymbol{z}_t - \boldsymbol{z}_t \boldsymbol{z}_t^\top \boldsymbol{\theta}\right] \\
&= -\sum_{t=1}^{T} x_t \boldsymbol{\omega}_t + \boldsymbol{\Omega}_t \boldsymbol{\theta}
\end{aligned}$$

$$\boxed{\boldsymbol{\theta}^* = \left(\sum_{t=1}^{T}\boldsymbol{\Omega}_t\right)^{-1}\left(\sum_{t=1}^{T} x_t \boldsymbol{\omega}_t\right)}$$

Problem 6: *Recurrent Neural Networks (RNNs) (20 pts)*

Consider a *linear* RNN that consumes a sequence of inputs $\boldsymbol{x} = (x_1, \ldots, x_L)$ with $x_\ell \in \mathbb{R}$ and produces a single output at the end, $y \in \mathbb{R}$. The log likelihood is,

$$p(y \mid \boldsymbol{x}) = \mathrm{N}(y \mid h_{L+1}, 1)$$

where the hidden states $h_\ell \in \mathbb{R}$ obey deterministic, linear dynamics for $\ell = 0, \ldots, L$,

$$h_{\ell+1} = wh_\ell + bx_\ell.$$

Assume $h_0 = 0$. The model has three real-valued parameters w, b, and x_0.

(a) (5 pts) Write the model as a linear regression,

$$p(y \mid \boldsymbol{x}) = \mathrm{N}(y \mid \theta_0 + \boldsymbol{\theta}^\top \boldsymbol{x}, 1)$$

where $\theta_0 \in \mathbb{R}$ is a scalar intercept and $\boldsymbol{\theta} \in \mathbb{R}^L$ is a vector of weights. Give an expression for θ_0 and the entries of $\boldsymbol{\theta}$.

(b) (5 pts) Plot the weights θ_ℓ as a function of ℓ for $\ell = 0, 1, \ldots, L = 20$ assuming $w = e^{-1/20} \approx 0.95$, $b = 1$, and $x_0 = 1$. Label your axes and key points like the y-intercept.

(c) (5 pts) Compute the derivative of the log likelihood with respect to the initial condition x_0.

(d) (5 pts) Why is the solution to part (c) problematic for learning x_0 with simple gradient descent?

Solution:

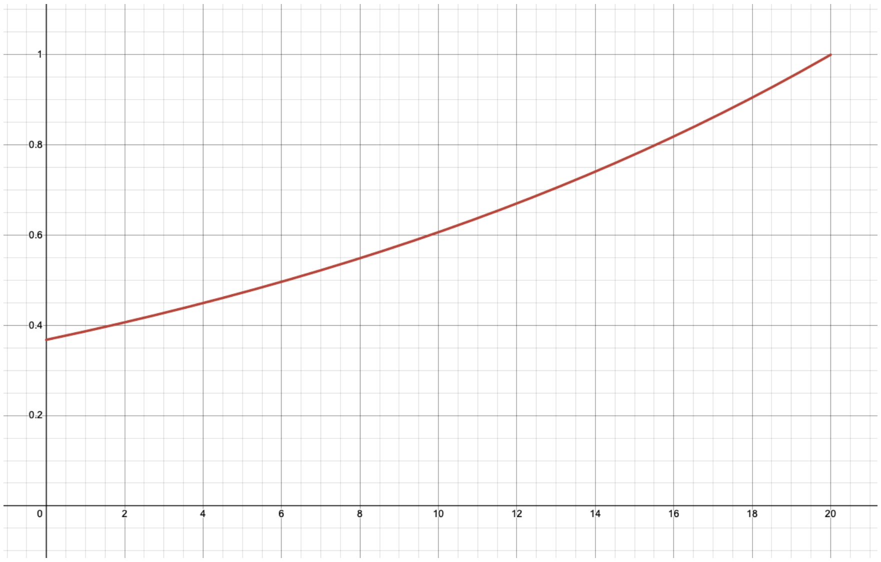
(a) We can write out the formula for some of the h_ℓs to look for a pattern:

$$\begin{aligned}
h_0 &= 0 \\
h_1 &= bx_0 \\
h_2 &= w + wbx_0 + bx_1 \\
&\vdots \\
h_t &= w^{t-1}bx_0 + \sum_{i=1}^{t-1} w^{t-1-i}bx_i
\end{aligned}$$

$$p(y|\boldsymbol{x}) = N\left(y \middle| w^L bx_0 + \sum_{i=1}^{L} w^{L-i}bx_i, 1\right)$$

Then $\boxed{\theta_0 = w^L bx_0}$ and $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_L)^\top$ where $\boxed{\theta_\ell = w^{L-\ell}b}$

(b)



The y-intercept is e^{-1}

(c)

$$\mathscr{L} = -\frac{1}{2}(y - w^L b x_0 - \boldsymbol{\theta}^\top \boldsymbol{x})^2 + C$$

$$\boxed{\frac{\partial}{\partial x_0}\mathscr{L} = -(y - w^L b x_0 - \boldsymbol{\theta}^\top \boldsymbol{x}) w^L b}$$

(d) When L is large, the gradient will vanish.

Problem 7: *Transformers (20 pts)*

Instead of the standard softmax attention, consider a more general form of *kernel* attention,

$$A_{t,s} = \frac{\mathrm{sim}(\boldsymbol{U}_q \boldsymbol{x}_t, \boldsymbol{U}_k \boldsymbol{x}_s)}{\sum_{s'=1}^{T} \mathrm{sim}(\boldsymbol{U}_q \boldsymbol{x}_t, \boldsymbol{U}_k \boldsymbol{x}_{s'})},$$

where $\boldsymbol{U}_q \boldsymbol{x}_t \in \mathbb{R}^K$ are the *queries* and $\boldsymbol{U}_k \boldsymbol{x}_s \in \mathbb{R}^K$ are the *keys*. Assume $K < D$. The function $\mathrm{sim}(\cdot, \cdot)$ is a *kernel* that measures similarity between two vectors. (Softmax attention corresponds to the exponential kernel, $\mathrm{sim}(\boldsymbol{a}, \boldsymbol{b}) = \exp\{\boldsymbol{a}^\top \boldsymbol{b}\}$.) Here, we will consider the quadratic kernel instead,

$$\mathrm{sim}(\boldsymbol{a}, \boldsymbol{b}) = (\boldsymbol{a}^\top \boldsymbol{b})^2.$$

(a) (4 pts) Show that with the quadratic kernel, $\mathrm{sim}(\boldsymbol{a}, \boldsymbol{b})$ can be written as an inner product $\phi(\boldsymbol{a})^\top \phi(\boldsymbol{b})$ where $\phi : \mathbb{R}^K \mapsto \mathbb{R}^P$ maps the vectors into a feature space of possibly higher dimension. *Hint: note that* $\mathrm{Tr}(\boldsymbol{A}^\top \boldsymbol{B}) = \mathrm{vec}(\boldsymbol{A})^\top \mathrm{vec}(\boldsymbol{B})$ *is an inner product.*

(b) (6 pts) In class, we wrote the output of a single-headed self-attention step as $\boldsymbol{Y} = \boldsymbol{A}\boldsymbol{X}$ where $\boldsymbol{X}, \boldsymbol{Y} \in \mathbb{R}^{T \times D}$ are matrices with rows $\boldsymbol{x}_t^\top$ and $\boldsymbol{y}_t^\top$, respectively, and $\boldsymbol{A} \in \mathbb{R}^{T \times T}$ is the matrix of attention weights. Show that with the quadratic kernel, the outputs matrix can be written as,

$$\boldsymbol{y}_t = \frac{\boldsymbol{V} \phi(\boldsymbol{U}_q \boldsymbol{x}_t)}{\boldsymbol{w}^\top \phi(\boldsymbol{U}_q \boldsymbol{x}_t)}$$

for some matrix $\boldsymbol{V} \in \mathbb{R}^{D \times P}$ and vector $\boldsymbol{w} \in \mathbb{R}^P$ that are the same for all outputs t. (Here, $\boldsymbol{x}_t$ and $\boldsymbol{y}_t$ are column vectors.)

(c) (4 pts) What is the computational complexity of the single headed self-attention step using the quadratic kernel? Your answer should be in terms of T, D, and/or K, but not P.

(d) (4 pts) What is the computation complexity of the single headed self-attention step using standard softmax attention? Your answer should be in terms of T, D, and/or K.

(e) (2 pts) In what regimes is quadratic attention more efficient? Do you expect Transformers to often be in this regime?

Solution:

(a)

$$\begin{aligned}
\mathrm{sim}(\boldsymbol{a}, \boldsymbol{b}) &= (\boldsymbol{a}^\top \boldsymbol{b})^2 \\
&= \mathrm{Tr}((\boldsymbol{a}^\top \boldsymbol{b})^2) \\
&= \mathrm{Tr}(\boldsymbol{a}^\top \boldsymbol{b}\boldsymbol{b}^\top \boldsymbol{a}) \\
&= \mathrm{Tr}(\boldsymbol{a}\boldsymbol{a}^\top \boldsymbol{b}\boldsymbol{b}^\top) \\
&= \mathrm{vec}(\boldsymbol{a}\boldsymbol{a}^\top)^\top \mathrm{vec}(\boldsymbol{b}\boldsymbol{b}^\top)
\end{aligned}$$

Therefore $\boxed{\phi(\boldsymbol{a}) = \mathrm{vec}(\boldsymbol{a}\boldsymbol{a}^\top)}$

(b)

$$\begin{aligned}\boldsymbol{y}_t &= \sum_{s=1}^{T} A_{t,s}\boldsymbol{x}_s \\ &= \sum_s \boldsymbol{x}_s \frac{\phi(\boldsymbol{U}_q\boldsymbol{x}_t)^\top \phi(\boldsymbol{u}_k\boldsymbol{x}_s)}{\sum_{s'} \phi(\boldsymbol{U}_k\boldsymbol{x}_{s'}^\top \phi(\boldsymbol{U}_k\boldsymbol{x}_{s'})} \\ &= \sum_s \boldsymbol{x}_s \frac{\phi(\boldsymbol{U}_k\boldsymbol{x}_s)^\top \phi(\boldsymbol{U}_q\boldsymbol{x}_t)}{\left(\sum_{s'} \phi(\boldsymbol{U}_k\boldsymbol{x}_{s'})^\top\right) \phi(\boldsymbol{U}_q\boldsymbol{x}_t)}\end{aligned}$$

$$\boxed{V = \sum_s \boldsymbol{x}_s \phi(\boldsymbol{U}_k\boldsymbol{x}_s)^\top, \quad w = \sum_s \phi(\boldsymbol{U}_k\boldsymbol{x}_s)}$$

(c) Computing V costs $O(TDP)$ because sum over T outer products of a D dimensional vector and a P dimensional vector. Computing w costs $O(TP)$

To compute the $\boldsymbol{y}_t$s, it takes $O(DP)$ to multiply the numerator and $O(P)$ to compute the dot product in the denominator. Overall, the cost of computing V dominates so computing all $\boldsymbol{y}_t$s is $O(TDP) = \boxed{O(TDK^2)}$ because $\phi(\boldsymbol{a})$ is $K \times K$ matrix flattened into a K^2 dimensional vector.

(d) Recall the standard softmax attention

$$A_{t,s} = \frac{\exp\left\{(\boldsymbol{U}_q\boldsymbol{x}_t)^\top(\boldsymbol{U}_k\boldsymbol{x}_s)\right\}}{\sum_{s'}^{T} \exp\{(\boldsymbol{U}_q\boldsymbol{x}_t)^\top(\boldsymbol{U}_k\boldsymbol{x}_{s'})\}}$$

For each t the cost of computing the denominator and numerator or $O(TK)$ and $O(K)$ respectively. The total time to compute A is $O(T(TK+K)) = O(T^2K)$. To compute $\boldsymbol{Y}$, we multiply a $T \times T$ matrix and a $T \times D$ which costs $O(T^2D)$. The overall cost is $O(T^2(D+K)) = \boxed{O(T^2D)}$ because we assume $K < D$.

(e) Quadratic kernel is faster when $K^2 < T$. Quadratic attention is faster in the regime where embedding dimension K is much smaller than context size T.