April 2022 Practice Exam
STA414H1S/2104H1S

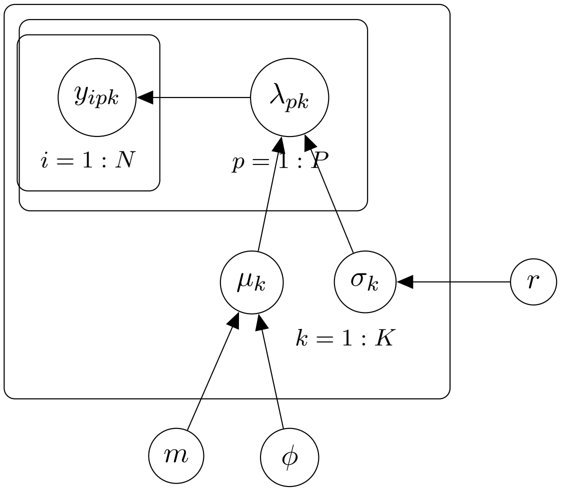
Statistical Methods for Machine Learning II

Duration - 170 minutes
Aids allowed: Two double-sided handwritten 8.5″ × 11″ or A4 aid sheets.

1. Translating from English to Graphical Models and Formulas. You are consulting for a research group in the sociology department investigating how arrest rates vary across different police officers based on a number of factors. They provide you with the following description of how they would like to model the dependencies:

- In the area of the country we are considering there are a total of K police divisions, split into P precincts each, with N police officers at each precinct.
- Each police officer will arrest some number of people based on the precinct directive average. We would like to model that as a Poisson distribution with the rate equal to that precinct average.
- Each precinct directive average is based on a division level average, and a division level variance, and normally distributed. We *a priori* assume that the division level variances are Exponentially distributed with rate of 10 arrests^2.
- We would like the division average to be modelled as normally distributed with a country-wide mean of 1000 arrests and a known variance.

1. Draw the graphical model representing the modeling described by your reasearch collaborators (5 points).
   Answer:



2. Write down the factorized joint distribution implied by the model described above, using the same names as in your graphical model. (5 points)
   Answer:

$$p(y_{1,1,1} \ldots, y_{N,P,K}, \lambda_{1,1}, \ldots, \lambda_{P,K}, \mu_1, \ldots, \mu_K, \sigma_1, \ldots, \sigma_K, \epsilon, \phi) =$$
$$= \prod_{k=1}^{K} \mathrm{Exp}(\sigma_k; r = 10) \mathcal{N}(\mu_k | m = 1000, \phi^2) \prod_{j=1}^{S} \mathcal{N}(\lambda_{pk} | \mu_k, \sigma_k^2) \prod_{i=1}^{N} \mathrm{Poiss}(y_{npk} | \lambda_{pk})$$

2. Probability and Sampling.

2.1. *Subsampling.* (4 points) Given a model whose likelihood factorizes over N datapoints:

$$p(x_1, \ldots, x_N|\theta) = \prod_{i=1}^{N} p(x_i|\theta)$$

write an unbiased estimator for $\log p(x_1, \ldots, x_N|\theta)$ that only uses one datapoint at a time.

Answer: $N \log p(x_i|\theta),$ where $i \sim U(1 \ldots N)$

2.2. *Rejection Sampling.* Imagine you want to sample from a complicated, but normalized distribution $p(x)$, and you know that it is upper-bounded by c times another distribution that you can sample from, $q(x)$.

- (5 points) Write pseudocode to sample from $p(x)$ using rejection sampling.

  Answer:
  ```
  do
      $x \sim q(x)$
      $u \sim U(0, 1)$
  while  $p(x) < (C \times u \times q(x))$
  return x
  ```

- What is the acceptance rate of your sampler?

  Answer: $\frac{1}{C}$

3. Markov Models, NLP, Neural Networks.

3.1. *Markov Models.*

1. You are training a Markov language model, in which the probability of the next token depends on only the k most recent tokens.
   For each of the following questions, what must k be in order for a model to be able to generate the follow sequences with 100% probability? (Each letter is a single token).

   (a) “ABABAB” Answer: 2

   (b) “AAAAAA” Answer: 0

   (c) “ABCABC” Answer: 1

   (d) “ABCABD” Answer: 3

2. Imagine you are given an autoregressive text model (such as a decoder only transformer), i.e. one that tells you $p(\text{next token}|\text{previous tokens})$. We want to use this model to “Fill in the blank“, i.e. evaluate the probability of n contiguous tokens in the middle of a sequence of length T, given all the rest of the tokens. There are K possible tokens in our vocabulary. For example, we might want to know what is the probability of seeing the sequence “A RED BALL” given only “A ___ BALL”. In which case, $n = 3$, $T = 10$, and $K = 27$ (for 26 letters and a space token).

   (a) (10 points) Derive an expression to compute $p(\text{missing tokens}|\text{observed tokens})$ exactly using only queries to $p(\text{next token}|\text{previous tokens})$. Denote the sequence of tokens $x_1, \dots, x_T$, and the indices of the start and end of the missing tokens as s and e, so that $p(\text{missing tokens}|\text{observed tokens}) = p(\underbrace{x_s, \dots, x_e}_{\text{middle}} \,|\, \underbrace{(x_1, \dots, x_{s-1})}_{\text{beginning}}, \underbrace{(x_{e+1}, \dots, x_T)}_{\text{end}})$,
   and the model gives us $p(x_{t+1}|x_1, \dots, x_t)$.
   Hint: Use the definition of conditional probability, which will require marginalizing over the missing tokens.

Answer:

(3.1)

$$\begin{aligned}
p(\text{missing tokens}|\text{observed tokens}) &= \frac{p(\text{missing tokens}, \text{observed tokens})}{p(\text{observed tokens})} \\
&= \frac{p(\text{missing tokens}, \text{observed tokens})}{\sum_{\text{missing tokens}} p(\text{missing tokens}, \text{observed tokens})} \\
&= \frac{p(\text{beginning}, \text{middle}, \text{end})}{\sum_{\text{middle}} p(\text{beginning}, \text{middle, end})} \\
&= \frac{p(\text{beginning})p(\text{middle}, \text{end}|\text{beginning})}{\sum_{\text{middle}} p(\text{beginning})p(\text{middle, end}|\text{beginning})} \\
&= \frac{p(\text{middle}, \text{end}|\text{beginning})}{\sum_{\text{middle}} p(\text{middle, end}|\text{beginning})} \\
&= \frac{\prod_{i=s}^{T} p(x_i|x_1, \ldots, x_{i-1})}{\sum_{x_s, \ldots, x_e} \prod_{i=s}^{T} p(x_i|x_1, \ldots, x_{i-1})}
\end{aligned}$$

(b) (3 points) How many evaluations of $p(\text{next token}|\text{previous tokens})$ does your method require? Answer: $(T-s)K^n$. For each of K^n possible sequences in the middle, you need to evaluate the model from every position from s to T. Note: You might put an exta "+ 1" for the numerator, either way is fine. Also acceptable is the slightly wasteful approach that doesn't cancel out the probabilities of the starting tokens, which has a time complexity of $TK^(T - s)$, as long as the two answers match.

3.2. *NLP / Attention.* Given access to an already trained Word2Vec model with embedding matrix U, your goal is to generate next word in a sequence based on the previous 2 words. The probability distribution over the words in your vocabulary conditional on previous 2 words is parametrized by:

$$p(x_i|x_{i-1}, x_{i-2}) = f(h_{i-2}, h_{i-1}; \theta) = softmax(Wz + b)$$

Where

$$z = \phi(h_{i-1}, h_{i-2})$$

with h_i, the Word2Vec embedding of the i-th token, and ϕ takes dimension-wise max for an array of vectors (so we aggregate across tokens).

1. (2 points) Assuming the embedding dimension for Word2Vec is 300, and the vocabulary size is K. What is the total number of parameters in this model, excluding the embedding matrix U?
   Answer: W has 300 x K, and b has 1 x K so a total of $301K$

2. (2 points) Imagine you also had access to a trained matrix of positional encodings P (2 x 300). Describe how you would normally add that information to the word embeddings, including how to obtain a value.
   Answer: You would sum up the Word2Vec embedding with the positional encoding, obtained by taking the first row of the matrix P for h_{i-2}, and the second row of the matrix P for h_{i-1}

Imagine instead of the vector z as defined above you used

$$z = \text{sum}(\text{Attention}(Q, K, V))$$

with Q, K, V each being $A \times 50$ matrices given by the learned linear projections W^Q, W^K, W^V of the 300-dimensional Word2Vec representation at each token.

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{300}}\right)V$$

denoting the scaled dot product attention mechanism, and the sum being taken over the dimension representing the sequence.

1. (4 points) What is the number of parameters in this model, excluding the Word2Vec embedding matrix?
   Answer: Each of the W^K,W^Q,W^V is 300 x 50, for a total of 3 x 300 x 50, plus the original matrix W in the final layer needs to be a 50 x K matrix, plus the b vector of size K for a grand total of 45,000 + 50K + 1K

2. (1 point) in the model above, what would A be ?
   Answer: 2, since we only have 2 token inputs

3. (2 points) If we wanted to extend this model to predict based on 5 previous words, what is the minimum number of parameters you would need to add to your model?
   Answer: 0, while Q,K,V would become larger, the projection matrices would still remain the same

4. (4 points) Suppose we replaced the learned projection matrices with the identity matrix of size 300 $W^Q = W^Q = \mathbb{I}$ (so that Q and K are A x 300). For an input token i, which token would the attention score be the highest for? Explain your reasoning. You may assume that there are no duplicate rows in the Word2Vec embedding matrix, and that all rows are normalized.
   Answer: It would be the highest for token i, the attention becomes:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{HH^\top}{\sqrt{300}}\right)V$$

Where H is the matrix of Word2Vec embeddings for both input tokens. We are guaranteed that the i-th score will be the highest because the dot product of the embedding will be the highest with itself since all rows are unique and normalized. (You can think of this as taking the cosine similarities between both the input word2vec embeddings and using their scaled pairwise similarities put through softmax as weights)

4. Gaussian Processes (8 points). in case they're useful, here are some properties of multivariate Gaussian vectors:

1. For a multivariate Gaussian vector $\mathbf{y} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ and a matrix $\boldsymbol{A}$, we have

$$\boldsymbol{A}\mathbf{y} \sim \mathcal{N}\big(\boldsymbol{A}\boldsymbol{\mu}, \boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{A}^T\big)$$

2. For any split,

$$\mathbf{y} = \begin{bmatrix} \mathbf{y}_A \\ \mathbf{y}_B \end{bmatrix} \sim \mathcal{N}\Big(\begin{bmatrix} \boldsymbol{\mu}_A \\ \boldsymbol{\mu}_B \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}_{AA} & \boldsymbol{\Sigma}_{AB} \\ \boldsymbol{\Sigma}_{BA} & \boldsymbol{\Sigma}_{BB} \end{bmatrix}\Big) \tag{4.1}$$

we have the conditional distribution again Gaussian

$$\mathbf{y}_A | (\mathbf{y}_B = \boldsymbol{a}) \ \sim \mathcal{N}\big(\boldsymbol{\mu}_A + \boldsymbol{\Sigma}_{AB}\boldsymbol{\Sigma}_{BB}^{-1}(\mathbf{x}_B - \boldsymbol{a}),\ \boldsymbol{\Sigma}_{AA} - \boldsymbol{\Sigma}_{AB}\boldsymbol{\Sigma}_{BB}^{-1}\boldsymbol{\Sigma}_{BA}\big). \tag{4.2}$$

The definition of a Gaussian process is: an infinite collection of random variables, such that any finite subset of them is jointly Gaussian distributed.

4.1. *Conjugacy.* (2 points) Imagine we put a Gaussian process prior on a player's skill at tennis at an particular time. So $z(t)$ denotes the player's skill at time t, and

$$z \sim \mathcal{GP}(\mu(t), k(t, t'))$$

Where z is a function $\mathcal{X} \to \mathbb{R}$, μ is a function $\mathcal{X} \to \mathbb{R}$, and k is a function $\mathcal{X} \times \mathcal{X} \to \mathbb{R}$

Further imagine that the speed of this player's serve is given by

$$p(\text{speed of serve at time } t | z(t)) = \mathcal{N}(\text{speed of serve at time } t | z(t), 10^2)$$

Is the true posterior over the player's skill at a particular time t_2, given an observation of their serve speed at time t_1, necessarily a Gaussian distribution? That is to say: is $p(z(t_2)|\text{speed of serve at time } t_1)$ Gaussian?

Answer: Yes. A Gaussian whose mean is given by another Gaussian variable forms a jointly multivariate Gaussian distribution. And any multivariate Gaussian conditioned on some of its values is always still Gaussian over the remaining variables.

4.2. *Bayesian Linear Regression.* (6 points)

Take a simple 1D Bayesian linear regression model, $p(w) = \mathcal{N}(w|0, I_D)$, and $y(x) = \sum_{d=1}^{D} w_i \phi_d(x)$, where x and y are real numbers, and $\phi(x)$ outputs a D-dimensional vector of features.

What is the marginal distribution of $y(x)$?

Answer: $\mathcal{N}(y(x)|0, \sum_{d=1}^{D}(\phi_d(x))^2)$. The sum of D independent normal variables has a mean that is the sum of all the variables' means, and a variance that is the sum of the variance of all the variables. Each $w_d\phi_d(x)$ is a Gaussian with mean zero and variance $(\phi_d(x))^2$.

5. Variational Inference (14 points).

5.1. *Reparametrization (6 points).* Suppose you have a function `icdf_t(p, nu)` which takes in a probability and outputs the inverse cumulative distribution function of a Student's t distribution with mean 0, variance 1, and degrees of freedom given by `nu`. You're also given a function `uniform_random(key)` which outputs a uniform random variable in $(0, 1)$ given a random seek `key`. Using this function, write pseudocode for a function that takes `mean` and `log_var`, the log of the desired standard distribution, and returns samples from a Student's t distribution with mean `mean`, variance `log_var`, and degrees of freedom `nu`:

```
def sample_students_t(key, mean, log_var):
```

Answer: `return mean + icdf_t(uniform_random(key), nu) * sqrt(exp(log_var))`

5.2. *True or false: (2 points).* For any particular model $P(x, z)$, (where x is discrete and z is continuous), discrete value of x, and family of approximate posteriors $q_\phi(z)$, it is always the case that:

$$\mathbb{E}_{q_\phi(z)}[\log P(x, z) - \log q_\phi(z)] < 0$$

Answer: True. The ELBO is a lower bound to $\log P(x)$, and since $P(x) \leq 1$, then

$$\text{ELBO} = \mathbb{E}_{q_\phi(z)}[\log P(x, z) - \log q_\phi(z)] \leq \log P(x) \leq \log 1 = 0$$

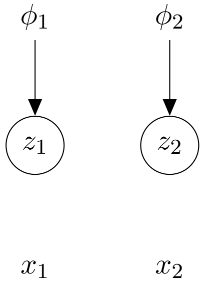
5.3. *Stochastic Variational Inference (6 points).* If we were to use plain stochastic variational inference in a latent variable model, we would have a separate set of variational paramters ϕ_i for the posterior over latents $q_{\phi_i}(z_i)$ corresponding to each datapoint x_i. So the joint variational posterior will have the following factorization:

$$q(z_1, \ldots, z_N) = \prod_{i=1}^{N} q_{\phi_i}(z_i)$$

1. (2 points) If we were to use a fully-factorized Gaussian approximate posterior for every $q_{\phi_i}(z_i)$, and each $z \in \mathbb{R}^{\mathbb{D}}$, how many variational parameters would we have in total for the

whole model? Answer: $2 \times N \times D$. For each of N datapoints, we'd have D dimensions of z_i, each of which would need one parameter for the mean and one for the variance.

2. (4 points) Draw the graphical model for the approximate posterior over 2 datapoints' latents (i.e. over $q(z_1, z_2 | \phi_1, \phi_2, x_1, x_2)$), with or without using plate notation, your choice. Reminder that variables that are not random can be drawn either with a shaded circle, or without a circle at all. The important thing is to make the dependencies and factorization clear. Answer:



5.4. *Bugs (2 points).* Imagine you are coding up a simple Monte Carlo estimator for the ELBO:

$$\text{ELBO}(\theta, \phi) = \mathbb{E}_{q_\phi(z)}[\log p_\theta(x, z) - \log q_\phi(z)]$$

and implement everything correctly, except you forget to use the reparameterization trick. For any given set of model parameters θ and variational parameters ϕ, will this code's simple Monte Carlo estimate of $\text{ELBO}(\theta, \phi)$ still be unbiased?

Answer: Yes. The reparameterization trick only affects estimation of the gradients of the ELBO wrt ϕ, not of the ELBO itself.