CPSC 532W - Homework 1

September 29, 2021

1. Assume the random variables $\mathbf{X} = \{X_1, X_2, \cdots, X_n\}$ follow the Poisson distribution, with the probability mass function:

$$f(\mathbf{X} \mid \lambda) = \prod_{i=1}^{n} \frac{\lambda^{x_i} e^{-\lambda}}{x_i!}$$

Let the prior be the Gamma distribution with parameters (α, β) :

$$p(\lambda \mid \alpha, \beta) = \frac{\beta^{\alpha}}{\Gamma(\alpha)} \lambda^{\alpha-1} e^{-\beta\lambda}$$

Then the posterior is

$$\begin{aligned} p(\lambda \mid \mathbf{X}, \alpha, \beta) &\propto f(\mathbf{X} \mid \lambda) p(\lambda \mid \alpha, \beta) \\ &= \prod_{i=1}^{n} \frac{\lambda^{x_i} e^{-\lambda}}{x_i!} \frac{\beta^{\alpha}}{\Gamma(\alpha)} \lambda^{\alpha-1} e^{-\beta\lambda} \\ &\propto \lambda^{(\sum_{i=1}^{n} x_i + \alpha - 1)} e^{(-(\beta+n)\lambda)} \end{aligned}$$

Therefore, the posterior follows Gamma distribution with parameters $(\sum_{i=1}^{n} x_i + \alpha, \beta + n)$, so the Gamma distribution is conjugate to the Poisson distribution.

2. First, we show the Gibbs transition operator satisfies the detailed balance equation. Assume we want to obtain a sample $\mathbf{x}'$ from $\mathbf{x} = (X_1 = x_1, \cdots, X_i = c, \cdots, X_n = x_n)$ from the joint distribution $p(\mathbf{x}) = p(x_1, \cdots, x_n)$ using Gibbs sampling. Suppose we randomly select the ith variable with probability π_i, then obtain $\mathbf{x}'$ by replacing value x_i with conditional probability

$$\Pr[X_i = c' \mid \mathbf{x}_{-i}] = \Pr[X_i = c' \mid x_1, x_2, \cdots, x_{i-1}, x_{i+1}, \cdots, x_n]$$

Therefore, the transition probability is

$$\Pr[\mathbf{x}' \mid \mathbf{x}] = \pi_i \Pr[X_i = c' \mid \mathbf{x}_{-i}]$$

and

$$\begin{aligned}
\Pr[\mathbf{x}' \mid \mathbf{x}]\, p(\mathbf{x}) &= \pi_i \Pr[X_i = c' \mid \mathbf{x}_{-i}]\, p(\mathbf{x}) \\
&= \pi_i \Pr[X_i = c' \mid \mathbf{x}_{-i}] \Pr[X_i = c \mid \mathbf{x}_{-i}]\, p(\mathbf{x}_{-i}) \\
&= \pi_i \Pr[X_i = c \mid \mathbf{x}_{-i}] \Pr[X_i = c' \mid \mathbf{x}_{-i}]\, p(\mathbf{x}_{-i}) \\
&= \pi_i \Pr[X_i = c \mid \mathbf{x}_{-i}]\, p(\mathbf{x}') \\
&= \Pr[\mathbf{x} \mid \mathbf{x}']\, p(\mathbf{x}') \\
\Rightarrow \Pr[\mathbf{x}' \mid \mathbf{x}]\, p(\mathbf{x}) &= \Pr[\mathbf{x} \mid \mathbf{x}']\, p(\mathbf{x}')
\end{aligned}$$

Here, we showed that the Gibbs transition operator satisfies the detailed balance equation.

To show this can be interpreted as an MH transition operator that always accepts, let $q(\mathbf{x}' \mid \mathbf{x})$ be the proposal distribution giving the probability of proposing $\mathbf{x}'$ to $\mathbf{x}$ and $\mathbf{x}'$ is differed by only one component, which is the transition probability $p(\mathbf{x}' \mid \mathbf{x})$ exactly. Since Metropolis-Hastings algorithm always accepts a proposed $\mathbf{x}'$ if

$$u \leq \frac{p(\mathbf{x}')q(\mathbf{x} \mid \mathbf{x}')}{p(\mathbf{x})q(\mathbf{x}' \mid \mathbf{x})}$$

where u is any random number uniformly generated between 0 and 1.
We have proved that the detailed balance equation is satisfied in this case by applying Gibbs sampling, we know that

$$p(\mathbf{x}')q(\mathbf{x} \mid \mathbf{x}') = p(\mathbf{x})q(\mathbf{x}' \mid \mathbf{x}) \Rightarrow \frac{p(\mathbf{x}')q(\mathbf{x} \mid \mathbf{x}')}{p(\mathbf{x})q(\mathbf{x}' \mid \mathbf{x})} = 1 \Rightarrow u \leq \frac{p(\mathbf{x}')q(\mathbf{x} \mid \mathbf{x}')}{p(\mathbf{x})q(\mathbf{x}' \mid \mathbf{x})} \quad \forall u$$

Therefore, for any generated number u between 0 and 1, MH algorithm will always accept the proposed $\mathbf{x}'$.

3. (a) The code and result are:

```
## condition and marginalize:
p_C_given_W = np.ones(2)
num = 0
denom = 0

for s in range(2):
    for r in range(2):
        num += p[1, s, r, 1]

for c in range(2):
    for s in range(2):
        for r in range(2):
            denom += p[c, s, r, 1]

p_C_given_W[1] = num/denom
p_C_given_W[0] = 1 - p_C_given_W[1]

print('There is a {:.2f}% chance it is cloudy given the grass is wet'.format(p_C_given_W[1]*100))
```

Figure 1: Code for enumerating all possible states

```
There is a 57.58% chance it is cloudy given the grass is wet
```

Figure 2: Result for enumerating all possible states

(b) The code and result is:

```
num_samples = 10000
samples = np.zeros(num_samples)
rejections = 0
i = 0
while i < num_samples:
    u1 = uniform(0,1)
    if (u1 >= p_C(0)) :
        c = 1
    else :
        c = 0
    u2 = uniform(0, 1)
    u3 = uniform(0, 1)
    if (u2 >= p_S_given_C(0, c)) :
        s = 1
    else :
        s = 0
    if (u3 >= p_R_given_C(0, c)) :
        r = 1
    else :
        r = 0

    u4 = uniform(0, 1)
    if (u4 >= p_W_given_S_R(0, s, r)) :
        w = 1
    else :
        w = 0
    if w == 1:
        if c == 1:
            samples[i] = 1
        i += 1
    else :
        rejections += 1
print('The chance of it being cloudy given the grass is wet is {:.2f}%'.format(samples.mean()*100))
print('{:.2f}% of the total samples were rejected'.format(100*rejections/(samples.shape[0]+rejections)))
```

Figure 3: Code for ancestral sampling and rejection

```
The chance of it being cloudy given the grass is wet is 57.72%
34.84% of the total samples were rejected
```

Figure 4: Result for ancestral sampling and rejection

(c) The code and result is :

```python
##gibbs sampling
num_samples = 10000
samples = np.zeros(num_samples)
state = np.zeros(4,dtype='int')
#c,s,r,w, set w = True
state[3] = 1
i = 0
while i < num_samples:
    u = randint(3, size = 1)
    if u == 0:
        u1 = uniform(0, 1)
        if u1 >= p_C_given_S_R[0, state[1], state[2]]:
            state[0] = 1
            samples[i] = 1

    elif u == 1:
        u1 = uniform(0, 1)
        if u1 >= p_S_given_C_R_W[0, state[0], state[2], state[3]]:
            state[1] = 1
            if state[0] == 1:
                samples[i] = 1

    elif u == 2:
        u1 = uniform(0, 1)
        if u1 >= p_R_given_C_S_W[0, state[0], state[1], state[3]]:
            state[2] = 1
            if state[0] == 1:
                samples[i] = 1

    i += 1

print('The chance of it being cloudy given the grass is wet is {:.2f}%'.format(samples.mean()*100))
```

Figure 5: Code for Gibbs sampling

```
The chance of it being cloudy given the grass is wet is 55.09%
```

Figure 6: Result for Gibbs sampling

4. (a) • Perform MH within Gibbs on the blocks $\mathbf{w}$: Let the proposal distribution be $q(\cdot)$, then if any generated number u from uniform distribution $\text{Unif}(0, 1)$ satifies

$$u \leq \frac{p(\mathbf{w}' \mid \mathbf{x}, \mathbf{t}, \sigma^2, \alpha) q(\mathbf{w} \mid \mathbf{w}')}{p(\mathbf{w} \mid \mathbf{x}, \mathbf{t}, \sigma^2, \alpha) q(\mathbf{w}' \mid \mathbf{w})}$$

we accept the proposed $\mathbf{w}'$ going from $\mathbf{w}$, and

$$
\begin{aligned}
r &= \frac{p(\mathbf{w}' \mid \mathbf{x}, \mathbf{t}, \sigma^2, \alpha) q(\mathbf{w} \mid \mathbf{w}')}{p(\mathbf{w} \mid \mathbf{x}, \mathbf{t}, \sigma^2, \alpha) q(\mathbf{w}' \mid \mathbf{w})} \\
&= \frac{p(\mathbf{t}, \mathbf{x}, \sigma^2, \mathbf{w}', \alpha) q(\mathbf{w} \mid \mathbf{w}')}{p(\mathbf{t}, \mathbf{x}, \sigma^2, \mathbf{w}, \alpha) q(\mathbf{w}' \mid \mathbf{w})} \\
&= \frac{\prod_{n=1}^{N} p(t_n \mid x_n, \sigma^2, \mathbf{w}') p(\mathbf{w}' \mid \alpha) q(\mathbf{w} \mid \mathbf{w}')}{\prod_{n=1}^{N} p(t_n \mid x_n, \sigma^2, \mathbf{w}) p(\mathbf{w} \mid \alpha) q(\mathbf{w}' \mid \mathbf{w})} \\
&= \frac{\prod_{n=1}^{N} \exp\left(-\frac{(t_n - \mathbf{w}'^\top x_n)^2}{2\sigma^2}\right) \exp\left(-\frac{1}{2} \mathbf{w}'^\top \left(\alpha \mathbf{I}_{d\times d}\right)^{-1} \mathbf{w}'\right) q(\mathbf{w} \mid \mathbf{w}')}{\prod_{n=1}^{N} \exp\left(-\frac{(t_n - \mathbf{w}^\top x_n)^2}{2\sigma^2}\right) \exp\left(-\frac{1}{2} \mathbf{w}^\top \left(\alpha \mathbf{I}_{d\times d}\right)^{-1} \mathbf{w}\right) q(\mathbf{w}' \mid \mathbf{w})} \\
&= \frac{\exp\left(-\frac{1}{2}(\mathbf{t} - \mathbf{x}\mathbf{w}')^\top (\sigma^2 \mathbf{I}_{N\times N})^{-1} (\mathbf{t} - \mathbf{x}\mathbf{w}') - \mathbf{w}'^\top \left(\alpha \mathbf{I}_{d\times d}\right)^{-1} \mathbf{w}'\right) q(\mathbf{w} \mid \mathbf{w}')}{\exp\left(-\frac{1}{2}(\mathbf{t} - \mathbf{x}\mathbf{w})^\top \left(\alpha \mathbf{I}_{N\times N}\right)^{-1} (\mathbf{t} - \mathbf{x}\mathbf{w}) - \mathbf{w}^\top \left(\alpha \mathbf{I}_{d\times d}\right)^{-1} \mathbf{w}\right) q(\mathbf{w}' \mid \mathbf{w})}
\end{aligned}
$$

If q is normally-distributed, then $q(\cdot \mid \cdot)$ can be eliminated from both numerator and denominator.

- Perform MH within Gibbs on t: Let the proposal distribution be $q(\cdot)$, the if any generated number u from the uniform distribution $\mathrm{Unif}(0, 1)$ satisfies

$$
u \leq \frac{p(\hat{t}' \mid \mathbf{w}, \hat{x}, \sigma^2, \alpha) q(\hat{t} \mid \hat{t}')}{p(\hat{t} \mid \mathbf{w}, \hat{x}, \sigma^2, \alpha) q(\hat{t}' \mid \hat{t})}
$$

we accept the proposed $\hat{\mathbf{t}}'$ going from $\hat{\mathbf{t}}$, and

$$
\begin{aligned}
r &= \frac{p(\hat{t}' \mid \mathbf{w}, \hat{x}, \sigma^2, \alpha) q(\hat{t} \mid \hat{t}')}{p(\hat{t} \mid \mathbf{w}, \hat{x}, \sigma^2, \alpha) q(\hat{t}' \mid \hat{t})} \\
&= \frac{p(\hat{t}', \mathbf{w}, \hat{x}, \sigma^2, \alpha) q(\hat{t} \mid \hat{t}')}{p(\hat{t}, \mathbf{w}, \hat{x}, \sigma^2, \alpha) q(\hat{t}' \mid \hat{t})} \\
&= \frac{p(\hat{t}' \mid \mathbf{w}, \hat{x}, \sigma^2, \alpha) q(\hat{t} \mid \hat{t}')}{p(\hat{t} \mid \mathbf{w}, \hat{x}, \sigma^2, \alpha) q(\hat{t}' \mid \hat{t})} \\
&= \frac{\exp\left(-\frac{1}{2\sigma^2}(\hat{t}' - \mathbf{w}^\top \hat{x})^2\right) q(t \mid \hat{t}')}{\exp\left(-\frac{1}{2\sigma^2}(\hat{t} - \mathbf{w}^\top \hat{x})^2\right) q(\hat{t}' \mid \hat{t})}
\end{aligned}
$$

If q is normally-distributed, then $q(\cdot \mid \cdot)$ can be eliminated from both numerator and denominator.

(b) Perform pure Gibbs on $\mathbf{w}$: we sample $\mathbf{w}'$ by sampling for the kth component in $\mathbf{w}$:

$$
\begin{aligned}
p(\mathbf{w}' \mid \mathbf{w}) &= p(w_k' \mid \mathbf{w}_{-k}) \\
&\propto p(w_k', \mathbf{w}_{-k}) \\
&= p(\mathbf{w}') \\
&= p(\mathbf{w}' \mid \mathbf{t}, \mathbf{x}, \sigma^2, \alpha) \\
&\propto \prod_{n=1}^{N} p(t_n \mid \mathbf{w}', x_n, \sigma^2) p(\mathbf{w}' \mid \alpha) \\
&\propto \exp\left(-\frac{1}{2}(\mathbf{t} - \mathbf{x}\mathbf{w}')^\top (\sigma^2 \mathbf{I}_{N\times N})^{-1} (\mathbf{t} - \mathbf{x}\mathbf{w}')\right) \exp\left(-\frac{1}{2} \mathbf{w}'^\top (\alpha \mathbf{I}_{d\times d})^{-1} \mathbf{w}'\right)
\end{aligned}
$$

By 4.4.1 theorem from Murphy's book, this follows multivariate normal distribution with parameters

$$\mathbf{\Sigma}^{-1} = (\alpha \mathbf{I}_{d\times d})^{-1} + \mathbf{x}^\top(\sigma^2 \mathbf{I}_{N\times N})^{-1}\mathbf{x}, \qquad \boldsymbol{\mu} = \mathbf{\Sigma}(\mathbf{x}^\top(\sigma^2 \mathbf{I}_{N\times N})^{-1}\mathbf{t})$$

Perform pure Gibbs on $\hat{t}$: we sample $\hat{\mathbf{t}}'$ by sampling of the kth component in $\hat{t}$:

$$\begin{aligned} p(\hat{t}' \mid \hat{t}) &= p(\hat{t}'_k \mid \hat{t}) \\ &\propto p(\hat{t}') \\ &= p(\hat{t}' \mid \mathbf{w}, \hat{x}, \sigma^2) \\ &\propto \exp\left(-\frac{1}{2\sigma^2}(\hat{t}' - \mathbf{w}^\top \hat{x})^2\right) \end{aligned}$$

This also follows normal distribution with parameters $N \sim (\mathbf{w}^\top \hat{x}, \sigma^2)$.

(c) The posterior distribution is

$$\begin{aligned} p(\hat{t} \mid \mathbf{t}, \hat{x}, \mathbf{x}, \sigma^2, \alpha) &= \int p(\mathbf{w} \mid \mathbf{t}, \mathbf{x}, \sigma^2, \alpha)\, p(\hat{t} \mid \hat{x}, \mathbf{w}, \sigma^2)\, d\mathbf{w} \\ &\propto \int \exp\left(-\frac{1}{2}(\mathbf{w} - \boldsymbol{\mu})^\top \mathbf{\Sigma}^{-1}(\mathbf{w} - \boldsymbol{\mu})\right) \exp\left(-\frac{1}{2\sigma^2}(\hat{t}' - \mathbf{w}^\top \hat{x})^2\right) d\mathbf{w} \end{aligned}$$

This is a convolution of two normal distributions, which still follows the normal distribution.

5. fstr1 'topic 0 :function, functions, linear, data, problem, set, basis, algorithm, kernel, method, topic 1 :data, distance, vectors, vector, space, algorithm, feature, set, points, dimensional, topic 2 :neurons, neuron, synaptic, spike, firing, model, time, cell, input, information, topic 3 :learning, algorithm, function, probability, theorem, examples, case, bound, set, number, topic 4 :state, time, model, control, states, algorithm, dynamic, space, step, optimal, topic 5 :network, networks, neural, function, input, output, layer, hidden, functions, units, topic 6 :features, set, feature, classifier, classification, recognition, object, training, rules, representation, topic 7 :model, motion, direction, system, visual, position, motor, eye, control, field, topic 8 :learning, dynamics, neural, function, networks, order, time, system, point, equation, topic 9 :model, cells, cortex, visual, cell, stimulus, input, response, activity, orientation, topic 10 :network, input, units, learning, output, unit, hidden, layer, weights, training, topic 11 :model, data, distribution, models, gaussian, probability, parameters, likelihood, mixture, bayesian, topic 12 :speech, recognition, word, training, system, context, neural, hmm, time, sequence, topic 13 :image, figure, images, graph, local, level, code, problem, model, vision, topic 14 :learning, reinforcement, policy, action, function, actions, system, task, time, reward, topic 15 :data, performance, number, results, set, search, neural, detection, test, human, topic 16 :neural, network, memory, neuron, networks, neurons, analog, input, weight, system, topic 17 :figure, signal, time, circuit, output, input, current, voltage, analog, frequency, topic 18 :training, error, data, set, learning, neural, networks, generalization, network, performance, topic 19 :images, information, image, face, component, analysis, independent, components, filter, source, '
fstr2 'Connectivity Versus Entropy , The Devil and the Network . , The Capacity of the Kanerva Associative

Memory is Exponential , Complexity of Finite Precision Neural Network Classifier , Single-iteration Threshold Hamming Networks , The Hopfield Model with Multi-Level Neurons , Shooting Craps in Search of an Optimal Strategy for Training Connectionist Pattern Classifiers , Examples of Learning Curves from a Modified VC-formalism , Performance Measures for Associative Memories that Learn and Forget , Worst-case Loss Bounds for Single Neurons , '