Problem 1.

1. $X \sim F^{-1}(u)$

$F(X) \sim F(F^{-1}(u)) \sim u$ ⟶ (A)

we want to show that $\mathbb{P}(F^{-1}(u) \le x) = F(x)$, $x \in \mathbb{R}$.

suppose: F is continuous, $\Rightarrow \{F^{-1}(u) \le x\} = \{u \le F(x)\}$.

so that: $\mathbb{P}(F^{-1}(u) \le x) = \mathbb{P}(u \le F(x)) = F(x)$. → $Unif(0,1)$.

$\Downarrow$

that is: $\mathbb{P}(Y \le x) = F(x)$, $Y \sim F^{-1}(u)$

$\Updownarrow$

Y is distributed as F, when $Y \sim F^{-1}(u)$. ■

Otherwise, we define $T: [0,1] \mapsto \mathbb{R}$ s.t. $T(u) \triangleq Y$

$F(x) = \mathbb{P}(Y \le x) = \mathbb{P}(T(u) \le x) = \mathbb{P}(u \le T^{-1}(x)) = T^{-1}(x)$

$\Leftrightarrow F = T^{-1}$, $\mathbb{P}(F^{-1}(u) \le x) = F(x) = \mathbb{P}(Y \le x)$.

Therefore, Y is distributed as F. ■

Drawback: When F does not have its inverse
we cannot use this method
Also, this method may be inefficient.

2. we already know that K_1 and K_2 have $p(\cdot)$.

$p(dy) = \int_G K_1(x, dy)\, p(dx)$, $y \in G$.

$p(dy) = \int_G K_2(x, dy)\, p(dx)$, $y \in G$.

we want to show that K_1K_2 and $\lambda K_1 + (1-\lambda)K_2$ also have $p(\cdot)$.

① $K_1 \circ K_2(x, dz) = \int K_2(x,y) K_1(y, dz)\, dy$

$$\int K_1 \circ K_2(x, dz) \cdot p(dx) = \iint K_2(x,y) K_1(y, dz)\, dy \cdot p(dx).$$

$$= \int \left(\int K_2(x,y)\, p(dx) \right) K_1(y, dz)\, dy.$$

$$= \int K_1(y, dz)\, p(dy) = p(dz)$$

Thus, we have $p(dz) = \int K_1 \circ K_2(x, dz) \cdot p(dx)$. ☒

② $\lambda p(dy) = \int_G \lambda K_1(x, dy)\, p(dx)$

$(1-\lambda) p(dy) = \int_G (1-\lambda) K_2(x, dy)\, p(dx)$

$\lambda p(dy) + (1-\lambda) p(dy) = p(dy) = \int_G [\lambda K_1 + (1-\lambda) K_2](x, dy)\, p(dx)$. ☒

3. Proof.

① the Markov chain with trasition distribution $T(x_{k+1} | x_k)$ imply that a stationary target distribution $p(x)$ will satisfy the equation $T(x_{k+1} | x_k) p(x_k) = T(x_k | x_{k+1}) p(x_{k+1})$.

That is how the Markov chain converge to the target.

② For MH sampling, when $x_{k+1} \neq x_k$, there must be an

accepted MH step with probability:

$$T(x_{k+1}|x_k) = \alpha(x_{k+1}, x_k)\, q(x_{k+1}|x_k)$$

when $x_{k+1} = x_k$, there may be

$$T(x_k|x_k) = \alpha(x_k, x_k)\, q(x_k|x_k) + \int_t q(t|x_k)(1-\alpha(t|x_k))\,dt$$

We know that, $T(x_{k+1}|x_k)p(x_k) = T(x_k|x_{k+1})p(x_{k+1})$ is easily satisfied with $x_{k+1} = x_k$, so we consider $x_{k+1} \neq x_k$.

$$T(x_{k+1}|x_k) = \min\left(1, \frac{p(x_{k+1})q(x_k|x_{k+1})}{p(x_k)q(x_{k+1}|x_k)}\right) q(x_{k+1}|x_k)$$

symmetric ↓

$$= \frac{1}{p(x_k)} \min\left(p(x_k)q(x_{k+1}|x_k),\ p(x_{k+1})q(x_k|x_{k+1})\right).$$

$$T(x_k|x_{k+1}) = \frac{1}{p(x_{k+1})} \min\left(p(x_{k+1})q(x_k|x_{k+1}),\ p(x_k)q(x_{k+1}|x_k)\right).$$

Thus, $p(x_k)T(x_{k+1}|x_k) = p(x_{k+1})T(x_k|x_{k+1})$, also satisfy. ☒

A continuous case is just an infinite adding of infinitesimal discrete cases, the symmetric still holds. ☒

4. Similarly, we want to show that the target distribution $p(x)$ of Gibbs sampling is the stationary distribution of the Markov chain, satisfying $T(x_{k+1}|x_k)p(x_k) = T(x_k|x_{k+1})p(x_{k+1})$.

We can prove this by proving that Gibbs sampling is a special case of MH sampling with samples being always accepted.

Proof. consider HM sampler as below:

$$q(\vec{x},\vec{y}) = q_i((x_1,\cdots,x_n),(x_1,\cdots,x,\cdots,x_n))$$

$$= \frac{1}{n} P(x_i = x \mid x_j = x_j, \forall j \neq i)$$

$$= \frac{1}{n} \frac{P(\vec{y})}{P(X_j = x_j, \forall j \neq i)}$$

$$\alpha(\vec{x},\vec{y}) = \min\left(1, \frac{f(\vec{y})q(\vec{y},\vec{x})}{f(\vec{x})q(\vec{x},\vec{y})}\right)$$

$$\Rightarrow \text{MH samples from } f(\vec{x}) = \begin{cases} 0 & \text{if } x \notin A \\ \frac{P(x)}{P(x \in A)} & \text{if } x \in A \end{cases}$$

Thus, for Gibbs sampler, if $\vec{x} \in A$ and $\vec{y} \in A$

$$\frac{f(\vec{y})q(\vec{y},\vec{x})}{f(\vec{x})q(\vec{x},\vec{y})} = \frac{\frac{P(\vec{y})}{P(\vec{y} \in A)} q(\vec{y},\vec{x})}{\frac{P(\vec{x})}{P(\vec{x} \in A)} q(\vec{x},\vec{y})} = \frac{P(\vec{y})q(\vec{y},\vec{x})}{P(\vec{x})q(\vec{x},\vec{y})} = \frac{\frac{P(\vec{y})}{P(\vec{x})}}{\frac{\frac{P(\vec{x})}{P(x_i = y_i)}}{\frac{P(\vec{y})}{P(x_j = x_j)}}}$$

$$= \frac{P(\vec{y})P(\vec{x})}{P(\vec{x})P(\vec{y})} = 1.$$

Also, if $\vec{x} \in A$, $\vec{y} \notin A$, $\frac{f(\vec{y})q(\vec{y},\vec{x})}{f(\vec{x})q(\vec{x},\vec{y})} = 0$. Thus, Gibbs sampler is a special MH sampler with $\alpha(\vec{x},\vec{y}) = \begin{cases} 1, & \vec{y} \in A \\ 0, & \vec{y} \notin A \end{cases}$

since we already prove the MH sampling will reach the target distribution of the stationary distribution of Markov chains, we know Gibbs sampling will also do that since under our setting, it is a special MH. ☒

Problem 2.

$$\begin{cases} y_1 = \sqrt{-2\log x_1}\cos 2\pi x_2 \\ y_2 = \sqrt{-2\log x_1}\sin 2\pi x_2 \end{cases}$$

$\Downarrow$

① $\dfrac{y_1}{y_2} = \dfrac{\cos 2\pi x_2}{\sin 2\pi x_2} = \dfrac{1}{\tan 2\pi x_2}$

$$\arctan\frac{y_2}{y_1} = 2\pi x_2, \Rightarrow x_2 = \frac{1}{2\pi}\arctan\frac{y_2}{y_1} = \frac{\theta}{2\pi}$$

② $y_1^2 + y_2^2 = -2\log x_1$

$$x_1 = \exp\left(\frac{y_1^2+y_2^2}{-2}\right). = \exp\left(\frac{r^2}{-2}\right)$$

Then it becomes $\begin{cases} y_1 = |r|\cos\theta \\ y_2 = |r|\sin\theta \end{cases}$

take $\dfrac{\partial(x_1,x_2)}{\partial(y_1,y_2)} = \begin{vmatrix} \frac{\partial x_1}{\partial y_1} & \frac{\partial x_1}{\partial y_2} \\ \frac{\partial x_2}{\partial y_1} & \frac{\partial x_2}{\partial y_2} \end{vmatrix}$

$$= \begin{vmatrix} \exp(-\frac{r^2}{2})\cdot\frac{2y_1}{-2} & \exp(\frac{r^2}{-2})\cdot\frac{2y_2}{-2} \\ \frac{1}{2\pi}\dfrac{-\frac{y_2}{y_1^2}}{1+(\frac{y_2}{y_1})^2} & \frac{1}{2\pi}\dfrac{\frac{y_1}{y_1^2}}{1+(\frac{y_2}{y_1})^2} \end{vmatrix}$$

$$= \exp\left(\frac{r^2}{-2}\right) \cdot \frac{1}{2\pi} \cdot \frac{1}{y_1^2+y_2^2} \underbrace{\left[(-y_1)\cdot y_1 - (-y_2)(-y_2)\right]}_{-y_1^2-y_2^2}$$

$$= \exp\left(\frac{r^2}{-2}\right) \cdot \frac{1}{2\pi} \cdot (-1)$$

$$= -\frac{1}{2\pi}\exp\left(-\frac{y_1^2+y_2^2}{2}\right)$$

$$= N(y_1|0,1)\,N(y_2|0,1) = \frac{\partial(x_1,x_2)}{\partial(y_1,y_2)} = p(y_1,y_2)$$

because: $dx_1 dx_2 = \frac{\partial(x_1,x_2)}{\partial(y_1,y_2)} dy_1 dy_2$

Therefore: $p(y_1,y_2) = N(y_1|0,1)\,N(y_2|0,1)$

$$= \int p(y_1,y_2|x_1,x_2)\,p(x_1,x_2)\,dx_1 dx_2$$

$$= \int p(y_1|x_1,x_2)\,p(y_2|x_1,x_2)\,p(x_1)\,p(x_2)\,dx_1 dx_2$$

Algorithm:

△ From above equations, we can see that sampling from norm distribution is equivalent to sampling from a circle: circle with r and θ, $r \sim \text{Unif}(0,1)$, $\theta \sim 2\pi \times \text{Unif}(0,1)$.

△ Therefore, we can design the algorithm as below:

① Sample x_1 and x_2 from $Unif(0,1)$.

② transform them into r and θ, using equations

$r=\sqrt{-2\log x_1}$ and $\theta=2\pi x_2$

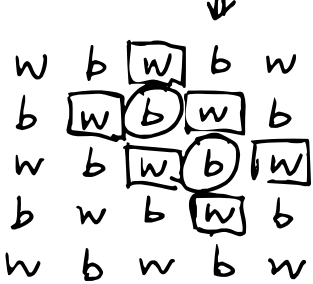
③ transform them into $\begin{cases} y_1=r\cos\theta \\ y_2=r\sin\theta \end{cases}$

△ If we only consider a univariate normal distribution we can fix the above result of $N(y_1|0,1)N(y_2|0,1)$ by fixing the value of y_1 (for example) to be a number, or we consider the case when $y_1=y_2$ by fixing the value of θ, then we have $N(y_1|0,1)N(y_1|0,1)$, this gives us the requried answer.

Problem 3.

△ Show the equation $p(b_1, \cdots | w_1, \cdots) = p(b_1 | w_1 \cdots)\, p(b_2 | w_1 \cdots) \cdots$

Proof: this is very intuitive. since all neighbours of white are black, and all neighbours of black are white if we condition on all white, the neighbours of all black are fixed, for example



(b) and (b) are independent since we know all [w] already

Therefore, all black are independent now, since this is Ising model, they are only dependent on their neighbours. Vice versa. ▨

△ Gibbs sampling: we perform the sampling using

① $p(b_1, b_2, \cdots | w_1, w_2, \cdots) = p(b_1 | w_1, w_2, \cdots)\, p(b_2 | w_1, w_2, \cdots) \cdots$

② $p(w_1, w_2, \cdots | b_1, b_2, \cdots) = p(w_1 | b_1, b_2, \cdots)\, p(w_2 | b_1, b_2, \cdots) \cdots$

we first independently sample from conditioned black points then we sample from condibioned white points.

↑↑ Gibbs sampling procedure.

Problem 4

Code as shown in problem4.py, I used a total iteration of 2500, a burning-in of 2000, and a sub-sampling of 20, the vector estimated is as below:

[7.205e-04 9.996e-01 1.650e-02 1.000e+00 5.841e-01 7.150e-03 8.114e-03
 7.976e-04 1.054e-03 1.000e+00 3.189e-03 1.000e+00 1.000e+00 1.000e+00
 9.843e-01 9.920e-01 9.989e-01 9.869e-01 1.000e+00 9.933e-01 4.589e-02
 8.556e-01 1.000e+00 9.836e-01 8.205e-03 9.998e-01 3.505e-03 5.139e-05
 9.979e-01 1.510e-08 5.898e-06 6.013e-02 4.243e-03 1.000e+00 4.684e-07
 4.331e-03 9.908e-01 1.111e-03 1.069e-06 5.954e-04 9.922e-01 9.976e-01
 1.000e+00 1.000e+00 4.733e-05 3.408e-03 9.997e-01 9.939e-01 3.916e-06
 9.998e-01]

Note that I chose a comparable low total iteration to save time, but this vector result already gives enough information of the disease situation: where some disease have a probability close to 1 and some disease have a probability close to 0.

Problem 5

collection of N patient records $D = (s^n, d^n)$

$$p(d|s,D) = p((d|s,D)|W,b,p)\, p(W,b,p)$$

$$= p(d|s,W,b,p)\, p(D|W,b,p)\, p(W,b,p)$$

because of independence →

$$p(D|W,b,p) = \prod_{n=1}^{N} p(s^n, d^n | W,b,p)$$

We already know that $p(\vec{s},\vec{d}) = \prod_{j=1}^{S} p(s_j|\vec{d}) \prod_{i=1}^{D} p(d_i)$

$$p(s_j|\vec{d}) = \sigma(w_j^T \vec{d} + b_j)$$

Thus, $\vec{s}$ is dependent on $\vec{d}$, W, b, while $\vec{d}$ dependant only on $\vec{p}$.

Thus, $p(s^n, d^n | W,b,p) = p(s^n | d^n, W, b)\, p(d^n | p)$

Combine them all:

$$p(d|s,D) = p(d|s,W,b,p)\, p(W,b,p) \prod_{n=1}^{N} p(s^n|d^n,W,b)\, p(d^n|p)$$

$$= \int_{W,b,p} p(d|s,W,b,p)\, p(W,b,p|D)$$ ∎

How to estimate $p(d_i=1 \mid s, D)$

Δ: use Gibbs sampling. → discribe as below.

First initialize $\vec{d}$ randomly.

For n in N iteration do

$$prob\_1 = p(d_i=1 \mid s, D)$$

$$\propto p(d_i=1 \mid s, w, b, p)\, p(w, b, p) \prod_{n=1}^{N} p(s^n \mid d^n, w, d)\, p(d^n \mid p)$$

$$prob\_0 = p(d_i=0 \mid s, D)$$

$$\propto p(d_i=0 \mid s, w, b, p)\, p(w, b, p) \prod_{n=1}^{N} p(s^n \mid d^n, w, d)\, p(d^n \mid p)$$

$$prob = \frac{prob\_1}{prob\_1 + prob\_0}$$ ⇐ so we can actually neglect a few parameters

↓

get new d_i according to prob and repeat for all d_i in d.

Problem 6

Code as shown in problem6.py, for problem (a), there is function called “problem_a” that generate 100 samples from 0.5N(-5,1)+0.5N(5,1). However, the samples used in problem (b) and (c) used the function “generate_samples”, which assume a different distribution.

Below are the results for problem (b) and (c):

Difference between sigma=0.5 and sigma=5: When sigma=5, the acceptance rate becomes significantly lower, and thus the number of samples accepted is significantly reduced, as shown in the second column of the below figure.

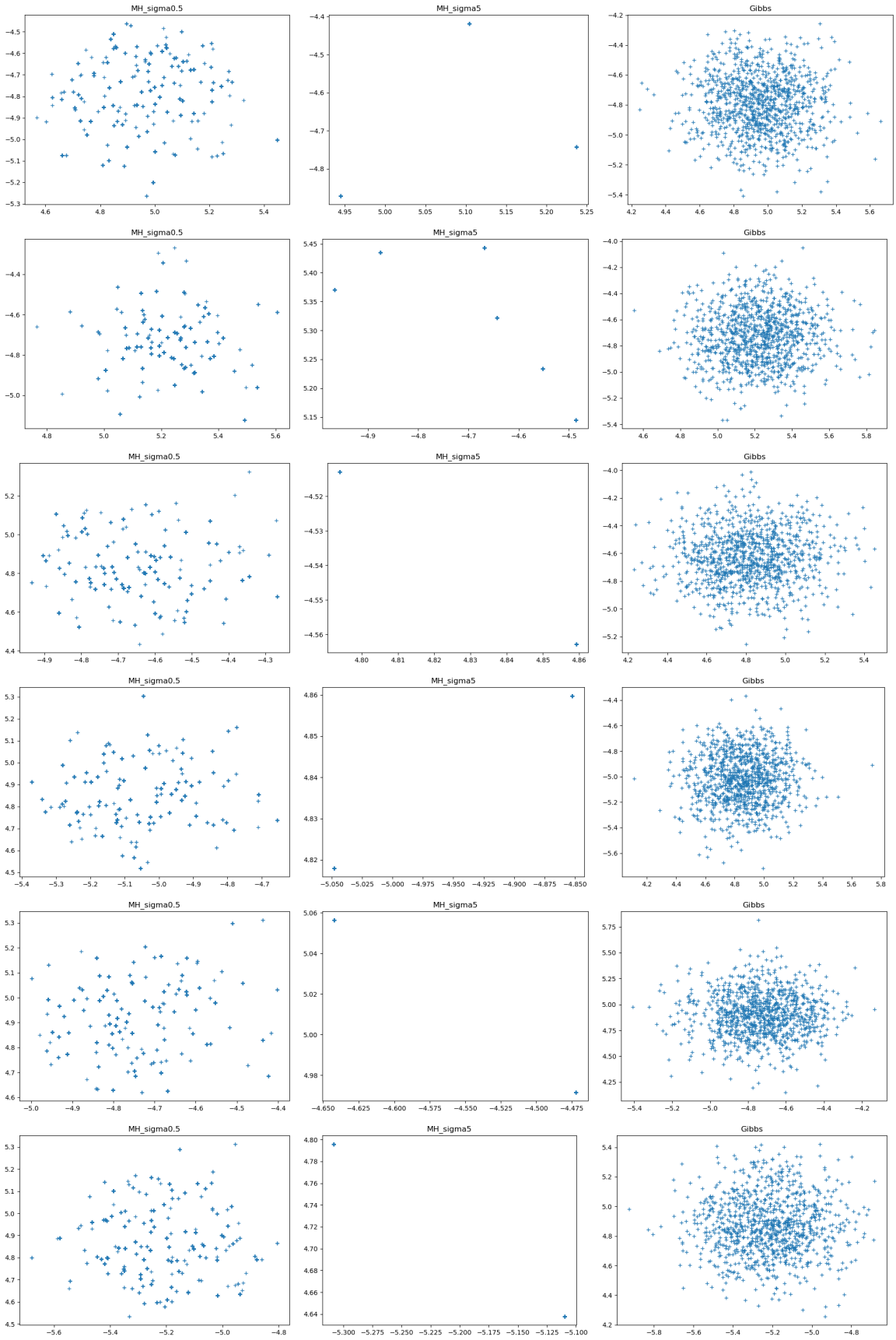
When sigma=0.5, the acceptance rate is around 0.12-0.13, while when sigma=5, the acceptance rate is around 0.002-0.003. This can be shown in figure.

sigma=0.5
ave_mu1: 4.964206507874 -- ave_mu2: -4.78491736293 -- accept_rate: 0.1370909090909
ave_mu1: 5.227456275805 -- ave_mu2: -4.73754133406 -- accept_rate: 0.12245454545454
ave_mu1: -4.652831786200 -- ave_mu2: 4.817549543821 -- accept_rate: 0.12418181818181
ave_mu1: -5.055239625157 -- ave_mu2: 4.851178611110 -- accept_rate: 0.13063636363636
ave_mu1: -4.748577779699 -- ave_mu2: 4.918870179142 -- accept_rate: 0.12263636363636
ave_mu1: -5.227551100533 -- ave_mu2: 4.866840581340 -- accept_rate: 0.1266363636363

sigma=5
ave_mu1: 5.09636272029 -- ave_mu2: -4.62277451665 -- accept_rate: 0.0024545454545454
ave_mu1: -4.745585728168 -- ave_mu2: 5.383749359624 -- accept_rate: 0.002272727272727
ave_mu1: 4.855203302291 -- ave_mu2: -4.5597718649523 -- accept_rate: 0.00209090909090
ave_mu1: -4.927526336550 -- ave_mu2: 4.843660716139 -- accept_rate: 0.002363636363636
ave_mu1: -5.234758374717 -- ave_mu2: 4.737384862023 -- accept_rate: 0.003
ave_mu1: -4.5746184250361 -- ave_mu2: 5.022751665995 -- accept_rate: 0.002

ave_mu1_G: 4.949819222114262 -- ave_mu2_G: -4.792380676849071
ave_mu1_G: 5.220048717732359 -- ave_mu2_G: -4.7274849674863875
ave_mu1_G: 4.820142475690202 -- ave_mu2_G: -4.62951098054149
ave_mu1_G: 4.856203978800593 -- ave_mu2_G: -5.02202178141987
ave_mu1_G: -4.73372303601228 -- ave_mu2_G: 4.901763283319164
ave_mu1_G: -5.219825178622551 -- ave_mu2_G: 4.870447429682438

For figures, the first column is sigma=0.5, second column is sigma=5, the third is Gibbs.


MH_sigma0.5
MH_sigma5
Gibbs
MH_sigma0.5
MH_sigma5
Gibbs
MH_sigma0.5
MH_sigma5
Gibbs
MH_sigma0.5
MH_sigma5
Gibbs
MH_sigma0.5
MH_sigma5
Gibbs
MH_sigma0.5
MH_sigma5
Gibbs

Problem 7

Code can be found in file p7, where the datafile is p.mat and code file is p7.m.

Code mimic the structure of a BRML demo:
https://github.com/taheris/BRML.jl/blob/d8897d0e3d0d2f72a055c67e2e55e875d0bd6868/matlab/DemosExercises/demoMFBPGibbs.m

I used the mean-field equations to optimize q.

Value of the minimal KL divergence is 0.229309.

Problem 8

Code can be found in file p8, where the p8.m is the main file, and the LBP_self.m is the Loopy belief propagation I wrote. Note that the structure of this function mimic the Loopy BP inside the BRML toolbox.

Code mimic the structure of a BRML demo (same as above):
https://github.com/taheris/BRML.jl/blob/d8897d0e3d0d2f72a055c67e2e55e875d0bd6868/matlab/DemosExercises/demoMFBPGibbs.m

----------LBP----------

var-(1)	var-(2)	var-(3)	var-(4)
0.0364	0.7064	0.4510	0.8302
0.9636	0.2936	0.5490	0.1698

----------MF----------

var-(1)	var-(2)	var-(3)	var-(4)
0.0021	0.9216	0.5668	0.8779
0.9979	0.0784	0.4332	0.1221

----------Exact----------

var-(1)	var-(2)	var-(3)	var-(4)
0.0325	0.7104	0.4509	0.8292
0.9675	0.2896	0.5491	0.1708

average error BP = 0.00218917
average error MF = 0.101594

Comments: BP works very well in this problem, with an average error of only 0.002. MF also gives an acceptable result but is worse than BP.

Problem 9.

$J = \log \int_x p(x) f(x)$ where $f(x) > 0$ is a positive function.

① $r(x) \propto p(x) f(x)$ is a distribution.

Thus, $\int_x r(x) = 1 \Leftrightarrow \int_x k p(x) f(x) = 1$.

$\Leftrightarrow \int_x p(x) f(x) = \frac{1}{k}$

$\Leftrightarrow J = \log \int_x p(x) f(x) = -\log k$

△ $D_{KL}(P \| Q) = \int_x p(x) \log \frac{p(x)}{q(x)} dx$ ← defination of KL.

$D_{KL}(q \| r) = \int_x q(x) \log \frac{q(x)}{r(x)} dx$

$= \int_x q(x) [\log q(x) - \log r(x)] dx$

$= \int_x q(x) \log q(x) dx - \int_x q(x) \log k p(x) f(x) dx$

$= \log k + \log p + \log f$

$= \langle \log q(x) \rangle_{q(x)} - \langle \log p(x) \rangle_{q(x)} - \langle \log f(x) \rangle_{q(x)} - \log k$

≥ 0

$\Leftrightarrow -\log k = J \geq \langle \log f(x) \rangle_{q(x)} + \langle \log p(x) \rangle_{q(x)} - \langle \log q(x) \rangle_{q(x)}$

we also know that

$$KL(q(x)|p(x)) = \int_x q(x) \log \frac{q(x)}{p(x)} dx$$

$$= \int_x q(x) [\log q(x) - \log p(x)] dx$$

$$= \langle \log q(x) \rangle_{q(x)} - \langle \log p(x) \rangle_{q(x)}$$

Thus: $J \geq \langle \log f(x) \rangle_{q(x)} - KL(q(x)|p(x))$ ▨

② we already have

$$J \geq \langle \log f(x) \rangle_{q(x)} + \langle \log p(x) \rangle_{q(x)} - \langle \log q(x) \rangle_{q(x)}$$

$$KL(q(x)|f(x)) = \langle \log q(x) \rangle_{q(x)} - \langle \log f(x) \rangle_{q(x)}$$

$$KL(q(x)|p(x)) = \langle \log q(x) \rangle_{q(x)} - \langle \log p(x) \rangle_{q(x)}$$

⇑ combining them

$$J \geq \langle \log f(x) \rangle_{q(x)} + \langle \log p(x) \rangle_{q(x)} - 2\langle \log q(x) \rangle_{q(x)} - \langle \log q(x) \rangle_{q(x)}$$

$$= \langle \log q(x) \rangle_{q(x)} - KL(q(x)|f(x)) - KL(q(x)|p(x))$$

$$= -\underbrace{H(q(x))}_{\text{entropy}} - KL(q(x)|f(x)) - KL(q(x)|p(x)).$$ ▨