05.11.21. Monte Carlo. Exam-Homework. Tokaeva Alexandra 509 gr.

1) We consider a non-negative function V on some finite set E,
$\beta > 0$ – some finite constant,
$M(x,y) = M(y,x)$ – some reversible Markov transition from E into itself.
We denote by $\pi(x)$ the Bolzmann-Gibbs probability measure

$$\pi_\beta(x) = \frac{1}{Z_\beta} e^{-\beta V(x)} \quad \text{with } Z_\beta = \sum_{y \in E} e^{-\beta V(y)}$$

We also consider an increasing sequence of parameters $\beta_n < \beta_{n+1}$ indexed by the discrete time index $n \in \mathbb{N}$.

1) Describe the Metropolis-Hastings algorithm with target measure π_β.

We are given target measure π_β, and we want to design a markov chain, that will have π_β as its stationary distribution.
This Markov chain will have the transition matrix $K(x,z)$, in other words, if we now ~~were~~ are in the state x, the next state of the chain will be z, obtained by the following rule $^{K(x,z)}$ (*):

1) Take x, and generate y from x, according to $M(x,y)$
2) Now we have our current state x, and proposed state y, and we need to decide, if we accept proposal (that is, decide, if $z = x$ or $z = y$).

We decide in the following way:

$$z = \begin{cases} y, \text{ with probability } a(x,y) \\ x, \text{ with probability } 1-a(x,y) \end{cases}, \text{ where } a(x,y) = \min\left(1; \frac{\pi(y) M(y,x)}{\pi(x) M(x,y)}\right)$$

What does this mean?

We have $\pi_\beta(x) = \frac{1}{Z_\beta} \cdot e^{-\beta V(x)}$

$$\Rightarrow a(x,y) = \min\left(1; \frac{\pi_\beta(y) M(y,x)}{\pi_\beta(x) M(x,y)}\right) = \min\left(1; \frac{\frac{1}{Z_\beta} \cdot e^{-\beta V(y)} M(y,x)}{\frac{1}{Z_\beta} \cdot e^{-\beta V(x)} M(x,y)}\right) = \min\left(1; e^{-\beta(V(y)-V(x))}\right) = e^{-\beta(V(y)-V(x))_+}.$$

So,

$$z = \begin{cases} y, \text{ if } V(y) \le V(x) \\ \text{if } V(y) > V(x), \text{ then } z = \begin{cases} y, \text{ with probability } e^{-\beta(V(y)-V(x))} \\ x, \text{ with probability } 1 - e^{-\beta(V(y)-V(x))} \end{cases} \end{cases}$$

Now, we have just made one step: from x to z.
Then make another step: take z as "current" state and rule (*) to give you "next" state.
And then go ahead in that way. After some amount of steps, when the chain converges, when you ask the chain to give another state – you will obtain elements from E according to the desired distribution π_β on E.

2) Describe a simulated annealing algorithm associated with the sequence of probability measures $(\pi_{\beta_n})_{n \in \mathbb{N}}$.

First take π_{β_1} as a target measure.

And run some amount of iterations of MCMC-algorithm (it is explained in a)) with target measure π_{β_1}. Stop when you feel that the chain has reached its stationary distribution (and it is π_{β_1} according to our construction of MCMC)

And now change your target measure from π_{β_1} to π_{β_2}.

Then again run some MCMC with target measure π_{β_2}.

Then change target measure from π_{β_2} to π_{β_3} and run some iterations of MCMC with π_{β_3}.

Then change target measure to the next $\pi_{\beta_\bullet}$, and run some MCMC — and so on.

3) Describe a genetic type algorithm based on a sequence of interacting simulated annealing moves.

First take π_{β_1} as target measure and simulate N independent MCMC chains with target measure π_{β_1}.

Then from each of N chains take the last element, let's call them $(\xi_1^i)_{i=1..N}$

And replace each ξ_1^i by $\hat{\xi}_1^i := \begin{cases} \xi_1^i, \text{ with probability } e^{-(\beta_2-\beta_1)V(\xi_1^i)} \\ \tilde{\xi}_1^i \text{ chosen from distribution } \sum_{k=1}^{N} \frac{e^{-(\beta_2-\beta_1)V(\xi_1^k)}}{\sum_{l=1}^{N} e^{-(\beta_2-\beta_1)V(\xi_1^l)}}\delta_{\xi_1^k}, \\ \text{with probability } 1-e^{-(\beta_2-\beta_1)V(\xi_1^i)}. \end{cases}$

~~Now you~~

Now you have "renewed" $(\hat{\xi}_1^i)_{i=1..N}$, and "next" target measure π_{β_2}, and you ~~[illegible]~~ run MCMC with target measure π_{β_2}, starting from each of $(\hat{\xi}_1^i)_{i=1..N}$. So you receive N chains. Take from each of them the last point. So you have N elements $(\xi_2^i)_{i=1..N}$.

⋮

After the n-th step, you receive N chains with last elements $(\xi_n^i)_{i=1..N}$.

So you take $(\xi_n^i)_{i=1..N}$, replace each ξ_n^i by $\hat{\xi}_n^i = \begin{cases} \xi_n^i, \text{ with proba } e^{-(\beta_{n+1}-\beta_n)V(\xi_n^i)} \\ \tilde{\xi}_n^i, \text{ with proba } 1-e^{-(\beta_{n+1}-\beta_n)V(\xi_n^i)}, \end{cases}$

where $\tilde{\xi}_n^i$ is chosen from distribution $\sum_{k=1}^{N} \frac{e^{-(\beta_{n+1}-\beta_n)V(\xi_n^k)}}{\sum_{l=1}^{N} e^{-(\beta_{n+1}-\beta_n)V(\xi_n^l)}}\delta_{\xi_n^k}$.

Then you take "renewed" $(\hat{\xi}_n^i)_{i=1..N}$, "next" target measure $\pi_{\beta_{n+1}}$, and again run N MCMC-chains, take N last elements, renew them, take next target measure, and so on.

2. We let λ be the probability measure of a random variable $W \sim N(0,1)$. $\forall x \in \mathbb{R}$, we let $M(x,dy)$ be the probability distribution of random variable $Y = \sqrt{1-\varepsilon}\,x + \sqrt{\varepsilon}\,W$.

We consider an interval $A = [a,b] \in \mathbb{R}$ with $a<b$, and we denote by $\pi(x)$ the probability measure
$$\pi(dx) = \frac{1}{Z_A} \mathbb{1}_A(x)\, \lambda(dx), \text{ with } Z_A = \int_{-\infty}^{+\infty} \mathbb{1}_A(y)\,\lambda(dy)$$

(2)/2

1) Check that $\pi(dx)M(x,dy) = \pi(dy)M(y,dx)$

$\pi(dx)$ has the density of $N(0,1)$ random variable

$\Rightarrow \pi(dx) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} dx$

$M(x;dy)$ has the density of random variable $y = \sqrt{1-e}\,x + \sqrt{e}\,W$,

where x is fixed, and is equal, and e is some preliminarily chosen constant, that doesn't change.

hence, $y \sim N(\sqrt{1-e}\,x; e)$.

$\Rightarrow M(x,dy)$ has the density of $N(\sqrt{1-e}\,x; e)$ random variable

$\Rightarrow M(x,dy) = \frac{1}{\sqrt{2\pi e}} e^{-\frac{(y-\sqrt{1-e}\,x)^2}{2e}} dy$

Similarly, $\pi(dy) = \frac{1}{\sqrt{2\pi}} \cdot e^{-\frac{y^2}{2}} dy$

$M(y;dx) = \frac{1}{\sqrt{2\pi e}} e^{-\frac{(x-\sqrt{1-e}\,y)^2}{2e}} dx$

So $\pi(dx)M(x,dy) \stackrel{?}{=} \pi(dy)M(y,dx)$ takes the form:

$$\frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} dx \cdot \frac{1}{\sqrt{2\pi e}} \cdot e^{-\frac{(y-\sqrt{1-e}\,x)^2}{2e}} dy \stackrel{?}{=} \frac{1}{\sqrt{2\pi}} e^{-\frac{y^2}{2}} dy \cdot \frac{1}{\sqrt{2\pi e}} e^{-\frac{(x-\sqrt{1-e}\,y)^2}{2e}} dx$$

$$\frac{1}{\sqrt{2\pi}} \cdot \frac{1}{\sqrt{2\pi e}} dx\,dy \cdot e^{-\frac{x^2}{2}} \cdot e^{-\frac{y^2}{2e}} \cdot e^{\frac{xy\sqrt{1-e}}{e}} \cdot e^{-\frac{(1-e)x^2}{2e}} \qquad \frac{1}{\sqrt{2\pi}} \cdot \frac{1}{\sqrt{2\pi e}} dx\,dy \cdot e^{-\frac{y^2}{2}} \cdot e^{-\frac{x^2}{2e}} \cdot e^{\frac{xy\sqrt{1-e}}{e}} \cdot e^{-\frac{(1-e)y^2}{2e}}$$

$$e^{-\frac{x^2}{2e}} \qquad e^{-\frac{y^2}{2e}}$$

$$\frac{1}{\sqrt{2\pi}} \frac{1}{\sqrt{2\pi e}} dx\,dy \cdot e^{-\frac{x^2}{2e}} e^{\frac{xy\sqrt{1-e}}{e}} \cdot e^{-\frac{y^2}{2e}}$$

So they are equal!

2) Describe the Metropolis-Hastings algorithm with target measure π.

When we run this algorithm, we will obtain the distribution $N(0,1) \mid [a,b]$.

What we do to obtain it?

We take any $x \in [a,b]$, and fix e.

Then ~~[illegible]~~ generate random $w \sim N(0,1)$,

and calculate $y := \sqrt{1-e}\,x + \sqrt{e}\,w$.

We have just made one step, from x to y.

Then take y as "current" state and generate w_{new}, and get next $y_{new} := \sqrt{1-e}\,y + \sqrt{e}\,w_{new}$

Then ~~[illegible]~~ make steps for some time, and soon you will obtain new points according to the distribution $N(0,1) \mid [a,b]$.

3) We consider a signal-observation filtering problem defined by

$$\begin{cases} X_n = a_n(X_{n-1}; W_n) \in \mathbb{R} \\ Y_n = h_n(X_n) + V_n \in \mathbb{R} \end{cases}, \text{ where } X_0; (W_n)_{n\geq 0}; (V_n)_{n\geq 0} \sim N(0,1), \text{ independent}$$

Functions $a_n: \mathbb{R}^2 \to \mathbb{R}$
$h_n: \mathbb{R} \to \mathbb{R}$ - regular.

a) Describe the particle filter algorithm to estimate the conditional distributions of X_n given the observations $Y_0 \dots Y_n$.

First take the distribution of $X_0 \sim N(0,1)$ and generate N samples $x_0^1 \dots x_0^N$ from it

~~[illegible]~~
These $x_0^1 \dots x_0^N$ stand for $p(x_0)$.

Then construct new distribution, that will aproximate $p(x_0/y_0)$, in that way:

~~[illegible]~~ the distribution takes point x_0^i with weight $\dfrac{p(y_0/x_0^i)}{\sum_{j=1}^{N} p(y_0/x_0^j)}$. $p(y_0/x_0^i)$ = density of $N(h_0(x_0^i); 1)$ random variable at point y_0

In other words, $p(x_0/y_0)dx_0 \approx \sum_{i=1}^{N} \left(\dfrac{p(y_0/x_0^i)}{\sum_{j=1}^{N} p(y_0/x_0^j)} \right) \delta_{x_0^i}(dx_0)$

Now you have "renewed" distribution, and generate N samples $(\hat{x}_0^1 \dots \hat{x}_0^N)$ from it

Then take each of $(\hat{x}_0^i)_{i=1\dots N}$, and calculate the corresponding x_1^i by the formula $X_n = a_n(X_{n-1}; W_n)$ (of course, you need to generate $W_n \sim N(0,1)$ for each x_0^i)

Now you have $(x_1^i)_{i=1\dots N}$ - they stand for $p(x_1/y_0)$

For them you now construct a new distribution ($\sim p(x_1/y_0; y_1)$), weighting the existing points $(x_1^i)_{i=1\dots N}$ with weights $p(y_1/x_1^i)$ = density of $N(h_1(x_1^i); 1)$ random variable at point y_1, because $Y_n = h_n(X_n) + V_n$.

Then you generate N "renewed" points $(\hat{x}_1^i)_{i=1\dots N}$ from this distribution.

Then you make transition $(\hat{x}_1^i)_{i=1\dots N} \rightsquigarrow (x_2^i)_{i=1\dots N}$, according to formula $X_n = a_n(X_{n-1}, W_n)$.

And so on.

That means, at the beginning of the n-th step, you have $(x_n^i)_{i=1\dots N}$ - they $\sim p(x_n/y_0 \dots y_{n-1})$.

You ~~[illegible]~~ obtain the approximation to distribution $p(x_n/y_0 \dots y_n)$ by weighting $(x_n^i)_{i=1\dots N}$ with weights $\dfrac{p(y_n/x_n^i)}{\sum_{j=1}^{N} p(y_n/x_n^j)}$

Then you sample $(\hat{x}_n^i)_{i=1\dots N}$ from that distribution, and ~~[illegible]~~ then calculate $(x_{n+1}^i)_{i=1\dots N}$ by the formula $X_{n+1} = a(X_n, W_{n+1})$. And so on

b) Describe the particle filter algorithm to estimate the conditional distributions of the trajectories $(X_0 \dots X_n)$ given the observations $Y_0 \dots Y_n$.

We do almost the same algorithm, as in 3a, but the "state" is ~~[illegible]~~ the tuple $(X_0 \dots X_n)$ instead of X_n, on the n-th step.

It means, that at the beginning of the n-th step, we have N tuples of length n. (c)p 3

They are $((x_0 \dots x_n)^i)_{i=1\dots N}$ – they stand for distribution of $(X_0 \dots X_n) / Y_0 = y_0 \dots Y_{n-1} = y_{n-1}$

Then we create new distribution, aproximating $(x_0 \dots x_n) / y_0 \dots y_n$, by weighting tuples $((x_0 \dots x_n)^i)_{i=1\dots N}$ with weights $p(y_n | (x_0 \dots x_n)^i) \underset{Y_n = h_n(X_n) + V_n}{=} p(y_n | x_n^i)$ = density of $N(h_n(x_n^i); 1)$ at point y_n.

Then generate $\widehat{(x_0 \dots x_n)}^i_{i=1\dots N}$ – N tuples according to this distribution on tuples

(It means, that each sample is a tuple of length n).

Then calculate $x^i_{n+1} = a_n(x^i_n; W_{n+1})$ for each x^i_n, and add x^i_{n+1} to the corresponding tuple

~~Then make next step~~ These N tuples $((x_0 \dots x_n x_{n+1})^i)_{i=1\dots N}$ are inputs for $(n+1)$-th step.

3) Describe an estimate of the density $p(y_0 \dots y_n)$ of the random variable $(Y_0 \dots Y_n)$

Let's denote $\gamma_n(f_n) := E\left(f_n(X_n) \cdot \prod_{0 \le k < n} G_k(X_k)\right)$, where $G_k(X_k) = p(y_k | x_k)$ – see algorithm in 3a.

Then $\prod_{0 \le k < n} G_k(X_k) = \prod_{0 \le k < n} p(y_k | x_k) \underset{y_k \text{ are independent when } x_k \text{ are fixed}}{=} p(y_0 \dots y_{n-1} / x_0 \dots x_{n-1})$.

Then $\gamma_n(1) = E\left(\prod_{0 \le k < n} G_k(X_k)\right) = E\, p(y_0 \dots y_{n-1} / x_0 \dots x_{n-1}) = \int\!\!\int\!\!\int_{x_0 \dots x_{n-1}} p(y_0 \dots y_{n-1} | x_0 \dots x_{n-1})\, dx_0 \dots dx_{n-1} = p(y_0 \dots y_{n-1})$.

So $\gamma_n(1) = p(y_0 \dots y_{n-1})$,

and $\gamma_{n+1}(1) = p(y_0 \dots y_n)$ – is what we need!

How to calculate $\gamma_n(1)$ sequentially, if we know $\gamma_{n-1}(1)$?

Obviously, $\gamma_n(1) = E\left(p(y_0 \dots y_{n-1} | x_0 \dots x_{n-1})\right) = E\Big(p(y_{n-1} | x_{n-1}) \cdot \underbrace{p(y_0 \dots y_{n-2} | x_0 \dots x_{n-2} \cancel{x_{n-1}})}_{\prod_{0 \le k < n-1} G_k(X_k)}\Big) \underset{y_k \text{ are independent}}{=} \gamma_{n-1}(p(y_{n-1} | x_{n-1}))$

So, $\gamma_n(1) = \gamma_{n-1}(p(y_n | x_n)) = \dfrac{\gamma_{n-1}(p(y_n | x_n))}{\gamma_{n-1}(1)} \gamma_{n-1}(1)$

How can we approximate $\dfrac{\gamma_{n-1}(p(y_n | x_n))}{\gamma_{n-1}(1)}$?

It is equal to $\dfrac{\int p(y_{n-1} | x_{n-1}) \prod_{0 \le k < n-1} dx_0 \dots dx_{n-1}}{\int \prod_{0 \le k < n-1} G_k(x_k)\, dx_0 \dots dx_{n-1}} \approx \dfrac{\int p(y_{n-1} | x_{n-1}) \cdot \sum \frac{1}{N} \cdot \delta_{x^i_{n-1}}(dx_0 \dots dx_{n-1})}{\underbrace{\int \sum_{i=1}^{N} \frac{1}{N} \delta_{x^i_{n-1}}(dx_0 \dots dx_{n-1})}_{=1}} = \dfrac{\sum_{i=1}^{N} p(y_{n-1} | x_{n-1}) \frac{1}{N}}{1}$

So $\frac{\gamma_{n-1}(p(y_{n-1}|x_{n-1}))}{\gamma_{n-1}(1)} \approx \frac{\sum_{i=1}^{N} p(y_{n-1}|x_{n-1}^{i})}{N}$.

But this sum – is the sum of weighs we calculate on the n-th step of algorithm 3a, when we construct a "renewed" weighted distribution.

Hence, $p(y_0 \dots y_n) = \gamma_{n+1}(1) = \prod_{k=1}^{n} \left(\frac{\sum_{i=1}^{N} p(y_k|x_k^i)}{N} \right)$ ◀

4. We consider a simple random walk $(X_n)_{n \in \mathbb{N}}$ on $\mathbb{Z}$ starting at the origin. We fix some parameter $L>0$ and we set $A=[-L;L]$.

1) Describe a particle algorithm to sample the conditional distribution of the random walk restricted to the set A.

▶ We have the same algorithm, as in 3a, but $G_k(x_k) = \mathbb{1}_{A}(x_k^i)$, instead of $p(y_k|x_k^i)$

So, at the beginning of n-th step, we have $(x_n^i)_{i=1\dots N}$.

We construct a new distribution, weighting x_n^i with weight $\mathbb{1}_{A}(x_n^i)$

Then we sample $(\hat{x}_n^i)_{i=1\dots N}$ from that distribution.

And for each $(\hat{x}_n^i)_{i=1\dots N}$, we calculate the corresponding $(x_{n+1}^i)_{i=1\dots N}$

Then $(x_{n+1}^i)_{i=1\dots N}$ are the inputs for the $(n+1)$-th step. ◀

2) Propose a sequential way to estimate of the probabilities $p_n := \mathbb{P}(X_0 \in A \dots X_n \in A)$.

▶ We have the same algorithm as in 3c, but $G_k(x_k^i) = \mathbb{1}_{A}(x_k^i)$ instead of $p(y_k|x_k^i)$

So $\gamma_n(1) = E\left(\prod_{0 \le k < n} G_k(X_k)\right) = E\left(\prod_{0 \le k < n} \mathbb{1}_A(X_k)\right) = \mathbb{P}(X_0 \in A; \dots X_n \in A)$.

And $\gamma_n(1) = \gamma_{n-1}(G_{n-1}(X_{n-1})) = \frac{\gamma_{n-1}(G_{n-1}(X_{n-1}))}{\gamma_{n-1}(1)} \gamma_{n-1}(1) \approx \sum_{i=1}^{N} \underbrace{G_n(x_n^i)}_{\mathbb{1}_A(x_n^i)} \cdot \gamma_{n-1}(1)$

Hence, $\mathbb{P}(X_0 \in A; \dots X_n \in A) = \gamma_{n+1}(1) = \prod_{k=1}^{n} \left(\frac{\sum_{i=1}^{N} \mathbb{1}_A(x_k^i)}{N} \right)$ ◀