FYS 5429, APRIL 19, 2023

- Markov chain Monte Carlo

p_i (model)

$$p_i(t) = \sum_j W(j \to i) p_j(t-1)$$

steady state

$$p(t=\infty) = W p(t=\infty)$$

$$p = Wp \qquad \lambda = 1$$

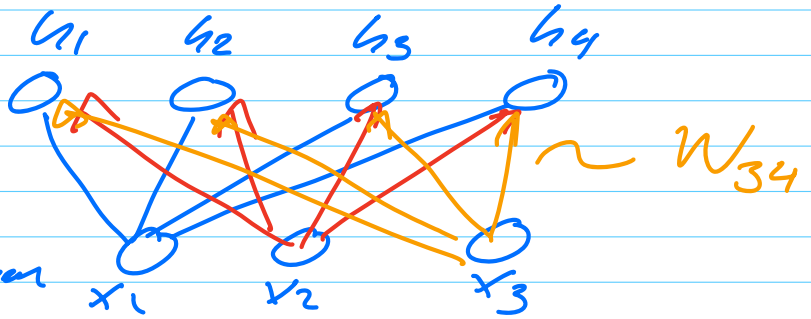
W is a stochastic matrix

$$\lambda_0 = 1 > \lambda_1 > \lambda_2 \cdots \lambda_{n-1}$$

$$p_i = \frac{e^{-\beta E_i}}{Z}$$

$$E(x, h) = -b^T x - c^T h - x^T W h$$

visible nodes ↑ ↖ hidden nodes



RBM = reduced Boltzmann

machine

Z = partition function

$$Z = \sum_i e^{-\beta E_i}$$

$$P_i \Rightarrow p(x,h) = e^{-\beta E(x,h)}$$

marginal probabilities

$$p(x|h) = \int_{h \in D} p(x,h)\, dh$$

$$\left(\sum_{h_i} p(x,h_i) \right)$$

$$p(h|x) = \int_{x \in B} p(x,h)\, dx$$

Sampling rules

- used when accepting new move/state/config

- $(MC)^2$
  - Metropolis-Hastings
  - Gibbs

$$W(j \Rightarrow i) = W_{ij}$$

$$\sum_j W_{i'j'} = 1$$

$$W_{i'j'} = T(j' \to i')\, A(j' \to i')$$

$T(j' \to i')$: transition probability

$A(j' \to i')$: acceptance probability

$$\frac{P_{i'}}{P_{j'}} = \frac{T(j' \to i')\, A(j \to i')}{T(i' \to j')\, A(i' \to j')}$$

$$T(j \to i') = T(i' \to j)$$

$$\frac{P_{i'}}{P_{j'}} = \frac{A(j \to i')}{A(i' \to j')} > 1$$

$0 \le A(j \to i') \le 1$

Metropolis-algo

$$A(j \to i') = \mathrm{Min}\left(1, \frac{P_{i'}}{P_{j'}}\right)$$

Metropolis-Hastings

$$A(j \to i') = \mathrm{Min}\left(1, \frac{P_{i'} T(i' \to j)}{P_{j'} T(j' \to i')}\right)$$

Gibbs - sampling.

We want k-samples of

$X = \{x_1, x_2, \ldots x_m\}$

from a distribution (Model)

$p(x_1, x_2, \ldots x_m) \mid p(x|h)$

We label the i'th sample as $X^{(i)} = \{x_1^{(i)}, x_2^{(i)}, \ldots x_m^{(i)}\}$

1) Begin with $k=0$, $X^{(0)}$ (random)

2) We want the next sample

$X^{(i+1)} = \{x_1^{(i+1)}, x_2^{(i+1)}, \ldots x_m^{(i+1)}\}$

$x_j^{(i+1)}$ is updated using the distribution

$P(x_j^{(i+1)} \mid x_1^{(i+1)}, x_2^{(i+1)}, \ldots x_{j-1}^{(i+1)}, x_{j+1}^{(i)} \ldots x_m^{(i)})$

3) Repeat k-times

Partition function (chap 18 of Goodfellow)

$$p(x,h) = \frac{e^{-\beta E(x,h)}}{Z}$$

$$p(x|h) = \int p(x,h)\,dh$$

$p(x;\theta)$

$\theta = \{b, c, W, \ldots\}$

$$p(x;\theta) = \frac{\tilde{p}(x;\theta)}{Z}$$

$$\int \tilde{p}(x)\,dx = Z = Z(\theta) = \sum_x \tilde{p}(x)$$

optimization of $\log p(x;\theta)$ wrt θ

$$\hat{\theta} = \arg\min_{\theta} \log p(x;\theta)$$

$$\nabla_\theta \log p(x;\theta) = \nabla_\theta \left[\log \tilde{p}(x;\theta) - \log Z(\theta)\right]$$

$e^{-\beta E(x;\theta)}$ = known

$$\nabla_\theta \log \tilde{p}(x;\theta) - \nabla_\theta \log Z(\theta)$$

positive phase of learning

negative phase

$$\nabla_\theta \log Z = \frac{\nabla_\theta Z}{Z}$$

$$= \nabla_\theta \frac{\sum_x \tilde{p}(x)}{Z} = \sum_x \frac{\nabla_\theta \tilde{p}(x)}{Z}$$

$\tilde{p}(x) > 0$ for all x

$$\sum_x \frac{\nabla_\theta \exp(\log \tilde{p}(x))}{Z}$$

$$= \sum_x \frac{\exp(\log \tilde{p}(x)) \nabla_\theta \log \tilde{p}(x)}{Z}$$

$$\sum_x \frac{\tilde{p}(x) \nabla_\theta \log \tilde{p}(x)}{Z} \sim p(x)$$

$=$

$$= \sum_x p(x) \nabla_\theta \log \tilde{p}(x)$$

$$= \mathbb{E}\left[\nabla_\theta \log \tilde{p}(x) \right]$$

$$\approx \frac{1}{N} \sum_i \nabla_\theta \log \tilde{p}(x_i)$$

$$\nabla_\theta \log p(x;\theta) =$$

$$\nabla_\theta \log \tilde{p}(x;\theta) -$$

$$\mathbb{E}\left[\nabla_\theta \log \tilde{p}(x;\theta) \right]$$

Algorithm MCMC for maximizing the log-likelihood use gradient ascent

Minimize → gradient descent

- set stepsize $\epsilon \approx 10^{-3}$ (learning rate)
- set of Gibbs steps

while not converged

- sample a minibatch $\{x_1, x_2 \ldots x_m\}$ from training set

$$g \leftarrow \frac{1}{m} \sum_{i=1}^{m} \nabla_\theta \log \tilde{p}(x_i; \theta)$$

- initialize a set of m samples $\{\tilde{x}_1, \tilde{x}_2, \ldots \tilde{x}_m\}$

for $i = 1$ to k

  for $j = 1$ to m

    $\tilde{x}_j \leftarrow$ gibbs_update$(\tilde{x}_j)$

  end

end

$$g \leftarrow g - \frac{1}{m} \sum_{i=1}^{m} \nabla_\theta \log \tilde{p}(\tilde{x}_i; \theta)$$

$\theta \leftarrow \theta + \varepsilon \cdot g$ / $(\theta - \varepsilon g)$

gradient descent.