Lecture FYS5429 April 2, 2024

FYS5429/9429 APRIL 2, 2024

iid = independent and identically distributed.

$$X = \{x_0, x_1, \ldots, x_{n-1}\}$$

$$P(X; \theta) = \prod_{i=0}^{n-1} p(x_i; \theta)$$

parameters

Example: Boltzmann distribution

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

$$Z = \int_{x\in D} dx \int d\theta \, e^{-\beta E(x;\theta)}$$

normalization constant

$$Z(\theta) = \int_{x\in D} dx \, e^{-\beta E(x;\theta)} \qquad \beta = \frac{1}{k_B T}$$

$$p(x) = \int d\theta \, e^{-\beta E(x;\theta)} = 1$$

if discrete variables

$$p(x) = \sum_{\theta_i} p(x;\theta_i)$$

$$Z(\theta) = \sum_{x_i \in D} p(x_i;\theta)$$

$E(x_i;\theta)$ energy function

(→ energy models)

$$E(x_i;\theta) \rightarrow E(x,h;\theta)$$

$$\theta = \{a, b, w\}$$

Binary - Binary model

$x_i = \{-1, 1\}$ $\quad$ $h_i = \{-1, 1\}$

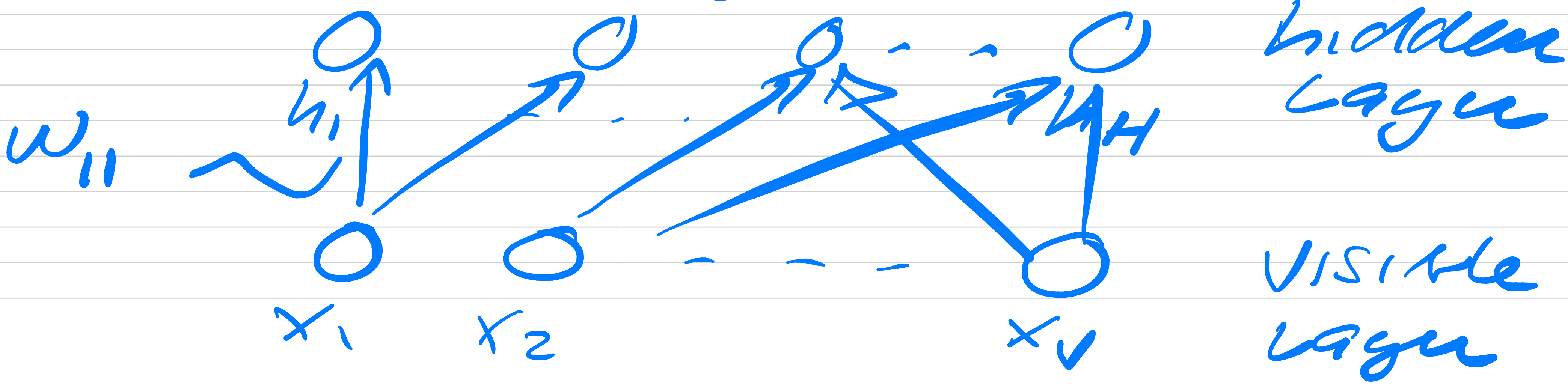
$E(x, h; \theta) =$

V = visible $\quad$ H = hidden

$$\sum_{i=1}^{V} a_i x_i + \sum_{j=1}^{H} b_j h_j + \sum_{i,j}^{VH} w_{ij} x_i h_j$$

bias

weights



Max likelihood

θ = Max likelihood Estimators

$$\arg\max_{\theta} P(x;\theta)$$

$$\arg\max_{\theta} \log P(x;\theta)$$

$$\Rightarrow \nabla_{\theta}\left(\sum_{x_i \in D} \log P(x_i;\theta)\right)$$

or

$$\arg\min_{\theta} \left(-\log p(x;\theta) \right)$$

$$p(x;\theta) = \frac{1}{Z(\theta)} \underbrace{f(x;\theta)}_{e^{-\beta E(x;\theta)}}$$

$$Z(\theta) = \int_{x \in D} f(x;\theta)\, dx$$

or

$$\sum_{x_i \in D} f(x_i;\theta)$$

we want to optimize $p(x;\theta)$ wrt θ

$$\nabla_\theta \log p(x;\theta) = \underbrace{\nabla_\theta \log f(x;\theta)}_{\text{positive phase}} - \underbrace{\nabla_\theta \log Z(\theta)}_{\text{negative phase}}$$

$$p(x;\theta) = \prod_{i \in D} p(x_i;\theta)$$

$$\log p(x;\theta) = \sum_{i \in D} \log p(x_i;\theta)$$

let us look at $\log Z(\theta)$

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

$$= \frac{\nabla_\theta \sum_{i \in D} f(x_i; \theta)}{Z(\theta)}$$

$$= \frac{\sum_{i \in D} \nabla_\theta f(x_i; \theta)}{Z(\theta)}$$

$p(x; \theta) \geq 0$

$\exp(\log(f(x_i; \theta)))$

$$= \frac{\sum_{i \in D} \nabla_\theta \left[\exp\{ \log(f(x_i;\theta)) \} \right]}{Z}$$

$$= \frac{\sum_{i \in D} \exp(\log(f(x_i;\theta))) \nabla_\theta \log f(x_i;\theta)}{Z}$$

$e^{-E(x_i;\theta)}$

$$= \sum_{i \in D} \frac{f(x_i;\theta) \nabla_\theta \log f(x_i;\theta)}{Z}$$

$\sim p(x_i;\theta)$

$$= \sum_{i \in D} p(x_i;\theta) \nabla_\theta \log(f(x_i;\theta))$$

$$= \mathbb{E}\left[\nabla_\theta \log(f(x_i;\theta)) \right]$$

$$\simeq \frac{1}{MCS} \sum_{j=1}^{MCS} \nabla_\theta \log(f(x_j;\theta))$$

$(MC)^2$ = # Monte carlo sampler

= Markov Chain Monte carlo

Markov Chain Monte Carlo

$P(x_i;t) \rightarrow P_i(t)$

$$\sum_{i \in D} P_i(t) = 1$$

Probability transition

$W_{ij} \rightarrow W(j \rightarrow i)$

$$P_i(t) = \sum_j W(j \rightarrow i) P_j(t-1)$$

depends only on $t-1$

$$\sum_j W_{ij} = 1$$ W is a stochastic matrix

largest eigenvalue $\lambda_0 = 1$

$$|\lambda_0| > |\lambda_1| > \dots > |\lambda_{m-1}|$$

eigenvectors v_i

$\alpha_i = \langle p_j | v_i \rangle$

$$p_j(t=0) = \sum_i \alpha_i v_i$$

when most likely state

is reached (steady state)

$P_i(t) = P_i(t-1)$ and

P_i is "time" independent

$P = WP$, eigenvalue problem with $\lambda = 1$, the largest eigenvalue of W

$$P(t=1) = \sum_i \alpha_i W v_i$$

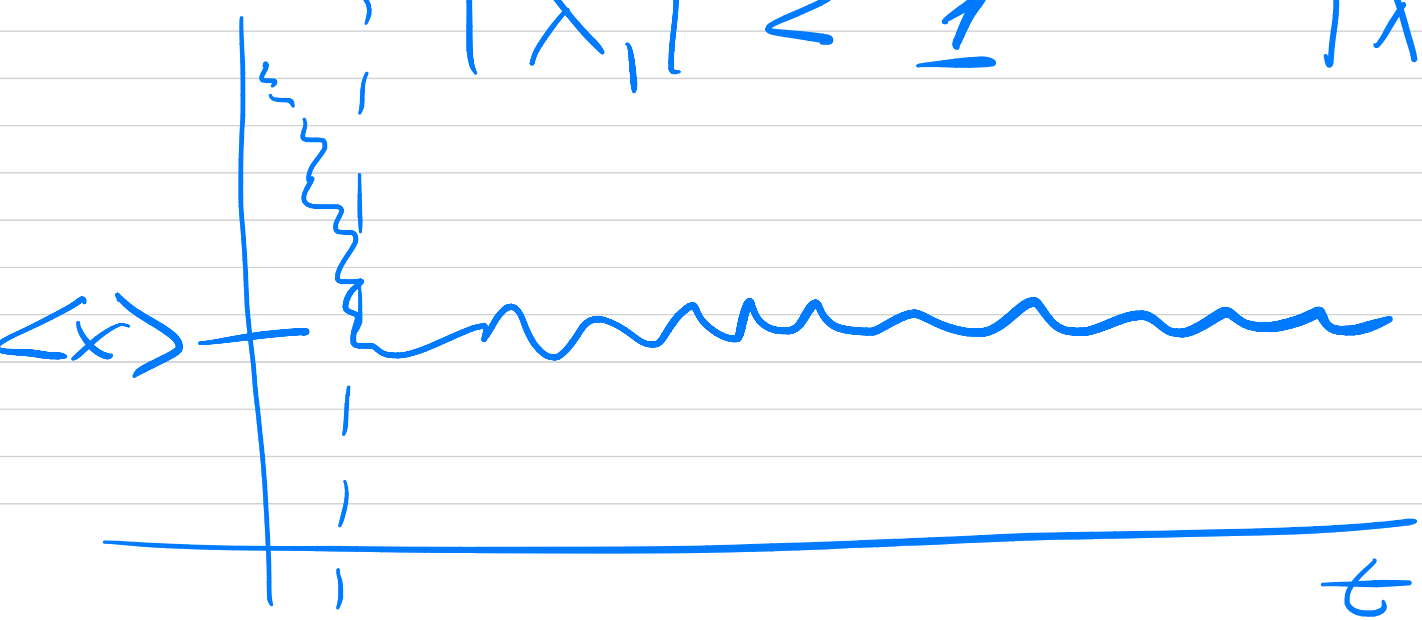
$$= \sum_i \lambda_i \alpha_i v_i$$

$$p(t) = W^t p(t=0)$$

$$= \sum_i \alpha_i \lambda_i^t v_i$$

$$= \underset{\substack{\| \\ 1}}{\lambda_0^t} \alpha_0 v_0 + \sum_{i \neq 0} \lambda_i^t \alpha_i v_i$$

$|\lambda_1| < 1 \qquad |\lambda_2| < |\lambda_1|$ etc



Two ways to accept a new value x_i

- Metropolis-Hastings-sampling
- Gibbs-sampling

Metropolis sampling

$$W(j \to i) = \underbrace{T(j \to i)}_{\text{transition probability}} \underbrace{A(j \to i)}_{\text{acceptance probability}}$$

Detailed balance

$$\frac{P_i}{P_j} = \frac{T(j \to i)\,A(j \to i)}{T(i \to j)\,A(i \to j)}$$

known

$$T(j \to i) = T(i \to j)$$

$$\frac{P_i}{P_j} = \frac{A(j \to i)}{A(i \to j)}$$

higher probability of staying in same place

$P_i / P_j \geq 1$

Metropolis algo

$$A(j \to i) = \min(1, P_i / P_j)$$

Gibbs sampling